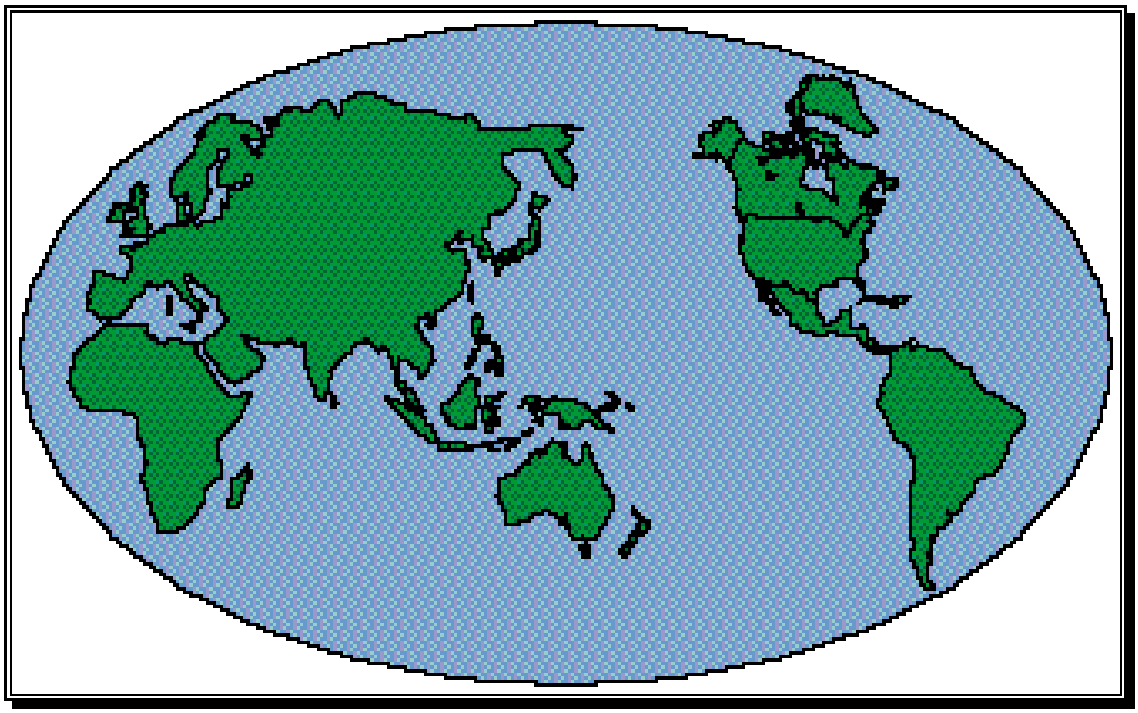
Introduction to HTML:

Web Page Authoring Basics at NIEHS



Edited by: Judi Fleming
Last updated: February 5, 1998
Information Technologies Support Services
National Institute of Environmental Health Sciences

Thanks to:

Roy Reter, Sarah Welna, Jed Dube, Sharon Hite and Sheila Beaudry for their input and assistance.

Basic outlines of this information can be found at:

http://www.niehs.nih.gov/websmith/

Note: **For the most current policy and procedures, always check the web site. As with all web-related items, this manual can become outdated very quickly. Do not assume that this is a timeless reference for creating web pages at NIEHS.**

Table of Contents

This manual is designed as an introduction to HTML and web page authoring. The objective is to give the learner a basic introduction to the concepts, tools and access to information on how to proceed to more advanced web page design on their own. It also lists resources of a more comprehensive nature as well as rules for creating web pages at NIEHS.

Appendix B

Appendix C

Chapter 1
Web Page Authoring Basics

Rules of the game:

I. If your branch or lab is part of DIR, you must get a DIR web account. If you are with any other division, you must get a Jeeves account. Either account can be applied for at: **http://www.niehs.nih.gov/websmith/restrict/jvsacnt.htm**.

To do web pages at NIEHS, you must be assigned web responsibilities for your Group, Branch/Lab, or Division.

1. Make the web page document (.htm), either typing the code onto a simple word processor (**SimpleText** for Mac; **NotePad** or **WordPad** for PCs) or by using the editor of your choice (**BBEdit 3.5** for Macintosh; **HomeSite** or **HTML Assistant Pro** for a PC).

   Also create the associated graphics (using only .jpg or .gif format *only*). You can use which ever software package is available to you. Check out the WebSmith's page for choices:
   http://www.niehs.nih.gov/websmith/webtools/imagview.htm

   a. Once the page is made, **save it** locally to your hard drive.
   b. Open Netscape and check your page by selecting **File**, **Open File**.
   c. Make changes as approriate.
   d. **Save** the file.
   e. Go back to Netscape and click on **Reload** to check the changes.

2. Upload each of these files to the appropriate directory space for your Group, Branch/Lab, or Division. (using an FTP client such as **Fetch** for the Macintosh; **WSFTP** or **FTP** on the MS DOS command line for PCs) or by connecting to the web servers through **Chooser** (Mac) or **Network Neighborhood** (PC).

   a. All web pages must be uploaded as **text** files, but have the **.htm** (prefered) or **.html** extension.
   b. All graphics must be uploaded as **raw data** and have the appropriate **.jpg** or **.gif** extension.

II. Announce that your page is live and make sure it is added to indexes and search engines, as approrpriate (send out an "all hands" message; email Marcia Soward so that your pages can be added to the NIEHS web page index at http://www.niehs.nih.gov/om/topics.htm).

Before You Begin

Before you begin authoring Web pages, you should be familiar with Netscape and the World Wide Web. You should be comfortable using Windows or the Macintosh operating system and have some word-processing experience. It's best if you also know how to manage your files, including how to move a file from one directory or folder to another. SCL offers classes and documents that give you the foundation you need to become a Web author. In this document you'll build on what you know to create and edit Web documents with or without an HTML editor such as BBEdit for the Macintosh or HomeSite for the PC. This first chapter only covers the first step of the "Rules of the Game" listed on the first page.

How To get your page on the Web at NIEHS

- Determine what you want to be displayed on the web
- Determine the html editor you are going to use
- Request an account on JEEVES or DIR (http://www.niehs.nih.gov/websmith/restrict/jvsacnt.htm)
- View NIEHS template requirements for html development
- Review supplemental HTML help documents

WWW publishing at NIEHS

At NIEHS, the main web server resides on **jeeves.niehs.nih.gov** and all of DIR web pages reside on **bedlam.niehs.nih.gov**. Access to Jeeves and the DIR Bedlam web servers are limited to persons who are in the People Database at NIEHS and who are responsible for creating web pages at NIEHS.

Creation of new accounts:

When requesting an account, you need to state the purpose of the page and which group at NIEHS it will represent.

Note: Personal, non-work related homepages are not supported but you can get a 30-day test account for computer class purposes.

Once your request is received, a new group directory will be created or an existing group directory will be modified, (which may already contain the files associated with your group at NIEHS). Included within this group are Jim Dix, Jed Dube and Sarah Welna. This allows these individuals to assist you with problems you might encounter while creating your web pages for those that fall under the Computer Technology Branch. Those individuals in DIR should request assistance from Roy Reter.

Initial page creation:

- The home page associated with your division, program, or group should be titled **home.htm**
- Files on the web server will be named in the "8.3" format. This means that the first part of the name should be eight characters or less, then the period, then the extension (.htm, .jpg, or .gif) should be three characters. Please note that the extension .html will work on most servers around the world, but not all of them.
- Three critical requirements to be added at the bottom of all web pages at NIEHS are:
    - contact person (with mailto link)
    - date of last update
    - URL
    - an organization name and a link out to another page

If you have any questions concerning this document or Web procedures and policies, please e-mail reter@niehs.nih.gov if you are within DIR or websmith@niehs.nih.gov if you are in any other division.

Prior to creating web pages

Acquire a Jeeves Account

To publish on the Web, you will need an account on Jeeves, which is a network file service here at NIEHS.

To obtain an account, go to:

http://www.niehs.nih.gov/websmith/restrict/jvsacnt.htm

to request an account, or have your supervisor email websmith@niehs.nih.gov. This account takes a minimum of two days to set up so that you have the proper directory assigned.

Build Directories

Once you have an account, you can begin to build your directories for your files. The initial page you create will be called **home.htm**. This will allow people to access your page by simply typing in the directory name.

For example: Under the websmith directory, there is a file called home.htm. To access the websmith homepage, simply type in http://www.niehs.nih.gov/websmith/

PC Users

When using a PC, you connect to Jeeves or DIR within the HTML folder. From there, you can locate your own folder. You may need to click on several folders to get to the folder you need to access.

Mac Users

When you open the Jeeves or DIR folder, you will be in the WWW folder. Double-click on the HTML folder, then double-click on the folder that contains your files. You may need to click on several folders to get to the folder you need to access.

Essentially, the directory from the root of Jeeves is:

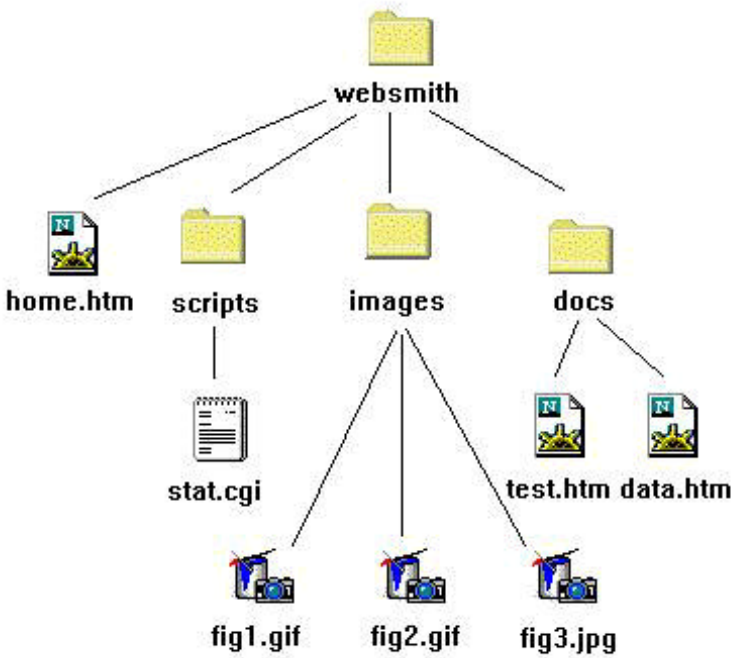
usr/local/WWW/html/*your group*

An example:

usr/local/WWW/html/websmith/docs

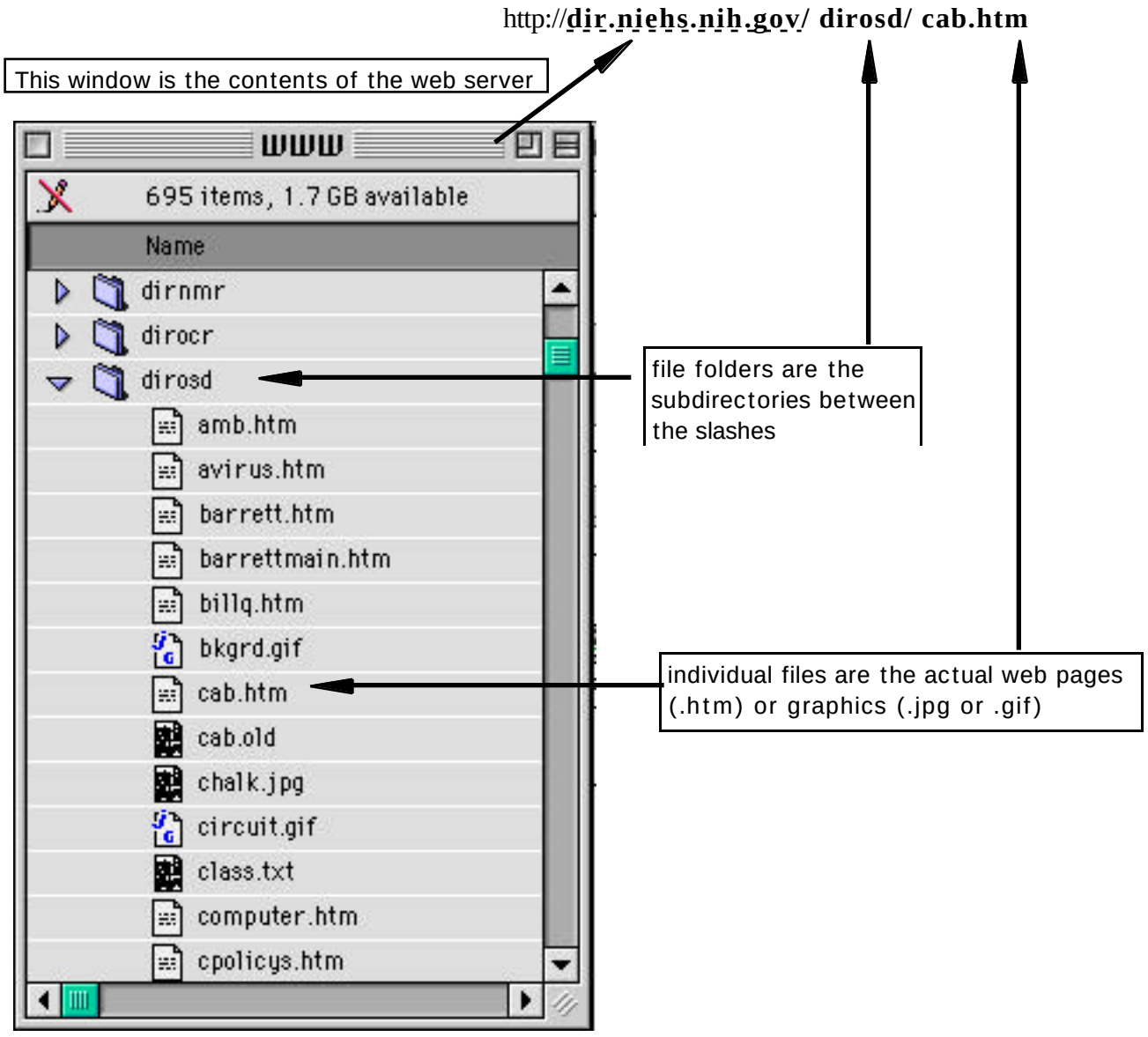
The home page (**home.htm**) appears as the web address:

http://www.niehs.nih.gov/websmith/



Basic Structure

Once you are in your directory, you will wan to have a basic layout that makes it easy for you to locate your files in the future. It is a good idea to have a directory to store all your image files. A basic file structure may look something like this:



Steps For Making Your Web Page

You'll get your Web page on-line and available to other people in a series of steps:

1. Plan your Web page. Who is your audience? What messages do you want to communicate? Sketch out your ideas *on paper*. Outlining skills help here. Spend the most time on this step. It will save you time and frustration later when you are re-doing pages.
2. Create a draft of your Web page.
3. Test your Web page locally from your computer desktop or file server. Use a graphical browser, like Netscape, as well as a text-only browser, like Lynx (if your page is to be an external page), to see how your page looks and works.
4. Put your Web page on the NIEHS server (using an FTP client)
5. Set the appropriate access privileges (permissions) for all of your Web files and directories.
6. Test your page with all available browsers (all versions of Netscape, Lynx, Mosaic an Explorer) and on all platforms (PC and Macintosh)--they can make your code look ***very*** different.
7. Revise your Web page as needed.
8. Notify other people that your Web page is open for business.

We'll talk about these steps more in this document, and we'll let you know when you need to refer to other chapters and helpful web pages for more in-depth treatment of some steps.

The first step is planning. This is probably the single most important step in web-page creation. Without a plan, you don't have a page. Or at best you'll have a sloppy page. Spending most of your time at this step will be your ultimate time saver. Sit down and put your ideas on paper just as if you were sketching a drawing before doing an actual painting. Brainstorm with your co-workers to find out what ideas they might have to make a richer, more interesting web page. Spend some time browsing the web for layout and designs that you find appealing and ask others to do the same.

DIR WEB PAGE INSTRUCTIONS FOR LAB CHIEFS AND GROUP LEADERS

Dr. Barrett has asked us to standardize and complete the DIR web page, which will describe intramural research projects at NIEHS. The information from these web pages will also be incorporated into a printed annaul report. Roy Reter, the DIR web master, will be carrying out this project. To facilitate his job we require each Lab Chief and Group Leader to supply the material described below.

- All text should be supplied in Word or Wordperfect using a 12 point Times font on a single disk or E-mailed as an attachment from each Lab Chief, including the text from all the Group leaders in that Laboratory.
- Individual photos (or an image of your preparation, technique or molecule of interest in its place) can be supplied as prints for scanning or as Photoshop files remembering that the final display size will be 1.5" square.
- Photos of individuals can be taken with a digital camera. Roy Reter should be contacted if such a service is needed.
- The graphical summary diagram, which each Group leader is required to submit, should be printed at final size (4" down, 6" across) for scanning and labelled clearly on the back.
- These graphical summary diagrams will be used to highlight your research.

Lab Chiefs should submit a list of the Research Groups and their leaders in addition to a short (4" down, 6" across) overview of the Laboratory's various research efforts and their relationship to environmental health.

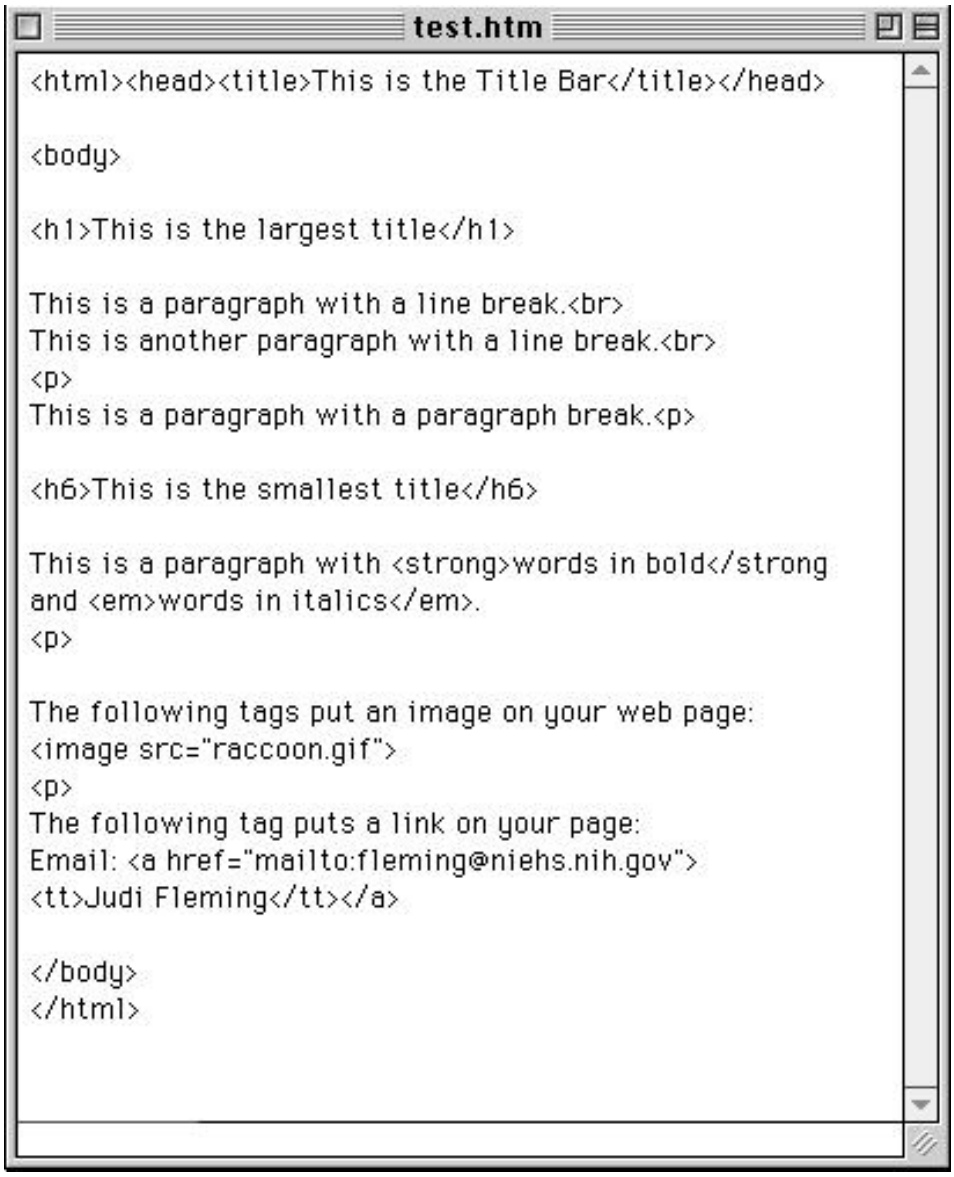
Lab Chiefs should also submit a list of one or two of the most important noteworthy achievements or advancements during the past year and their relevance to EHS. Principal Investigators (PIs) should indicate what subject (epidemiology, neuroscience, structural biology, etc.) best describes their work.

All research group leaders (including Lab Chiefs) should submit:

1. A picture of themselves or a relevant scientific image for display in a 1.5" square.
2. Bio info including
    - name, degree, date, Institution
    - members of group with Visiting scientists, postdocs, and technical staff listed on separate lines
3. Research summary of ~250 words or one page of text
4. Graphical research summary in a 4" down x 6" across box (legend included): this could be a diagram or a table or data
5. List of 5-10 references, most recent at the top
6. Address of any other web page to which you want a link provided

Submissions should be provided to Pat Magyar.

Once you've laid out your ideas for designs, then you must get down to the business of writing the HTML code and the text that will make up your page. HTML is a very simple set of tags and sort of looks like WordPerfect's code.



You can check you code as you go along without uploading it and making your page live. This allows you to change things on the fly and try things out without allowing the whole world to see your page while you are working on it. To do this, simply open Netscape and select **File Open** from the **File** menu item.

Style Guidelines for Web Pages at NIEHS

What makes a good Web page? The NIEHS Style Guidelines committee has some answers for you on their HTML Guidelines page at:

http://www.niehs.nih.gov/websmith/restrict/policy.htm

At NIEHS, the main web server resides on jeeves.niehs.nih.gov. As mentioned in previous sections, access to Jeeves is limited to persons who are in the People Database at NIEHS and who are responsible for creating web pages at NIEHS. Request a Jeeves account by going to: **http://www.niehs.nih.gov/websmith/restrict/jvsacnt.htm**.

Once your request is received, a new group will be created, or an existing group will be modified, with the files associated with your group at NIEHS.

Creation of new accounts:

When requesting an account, you need to state the purpose of the page and which group at NIEHS it will represent

- Personal, non-work related homepages are not supported
- Included within this group are Jim Dix, Jed Dube, Sarah Welna and the Websmith. This allows these individuals to assist you with problems you might encounter while creating your web pages.

Initial page creation:

1. The home page associated with your division, program, or group should be titled **home.htm**
2. Files on the web server will be named in the "8.3" format
3. Three critical requirements to be added at the bottom of all web pages at NIEHS are:
   - contact person
   - date of last update
   - an organization name and a link out to another page

If you have any questions concerning this document or Web procedures and policies, please e-mail

reter@niehs.nih.gov for those in DIR
websmith@niehs.nih.gov for those in any other division

We'll try to follow these guidelines as you learn to create your own Web page using this document and continuing with advanced Web page design in the chapters following this. But first let's look now at how to create your Web page now that we've got the administrative details out of the way.

Writing in the Language of Browsers: HTML

The basic tool for transforming text, graphics, and multi-media files into a completed Web page is the **Hypertext Markup Language** (**HTML**). Netscape, Mosaic, Lynx, and other browsers read HTML "tags." Tags are commands that tell browsers what to do, such as where to insert line breaks in text, where to show a link, or what file to display when a reader selects a link. To create a Web page, you put together text and HTML tags in a text file, called the "source" file. Web browsers interpret this source file, and display it on the Web. Each browser may interpret and display the source file a bit differently. In this document and in the corresponding class, we will be using Netscape as our primary browser. Here are some basic cautions

- The tags can be in any case or even mixed case
- Hitting the return key on the keyboard will not show up on the web page
- There are limitations to layout in web pages that you do not have in word processing software

Basics Format of HTML Tags

Generally, to give a browser a command, you sandwich text between two tags; a beginning tag turns the command on, and a corresponding end tag turns the command off. For example, you might want to make a heading for a paragraph, as in this example:

<h1>A Few Of My Favorite Things</h1>

The browser displays only the text between the tags <h1></h1> ("A Few Of My Favorite Things"), not the tags themselves. Notice that beginning tags are in the format **<tag>**, and end or closing tags are in the format **</tag>**. Some commands do not use end tags, as you'll see later in this document. HTML tags, by the way, are **not** case-sensitive, so you can type them in uppercase (ABC) or lower case (abc) or a mixture of either case (Abc). Also, some tags do not use their entire name in the end or closing part of the tag such as <a href> . The best way to learn to use tags is to put together a Web document. Let's begin one now.

If a Browser Does not Understand a Tag Will Something Terrible Happen?

When a browser doesn't understand a tag, it simply ignores it and goes on to the next one. For example, Lynx ignores any text color tags you use, as well as the special table formatting and other tags you'll learn about in this manual. If you don’t close a tag properly, that tags properties continue until it reaches a close tag for it or the end of the document. This can have some very humorous, if not frustrating, results.

Creating and Saving the File Simple Web Document

You can create your file with a simple word processor (like **Note Pad** for Windows or **Simple Text** for the Macintosh) or an online text editor (like **pico** or **ne** in UNIX). You can also use a more powerful word-processor than the ones we've mentioned or even a Web page editor, but it's not necessary. NIEHS is permitted to use BBEdit 3.5 for the Macintosh and has a site license for HomeSite for PC users. Whether you type the code out yourself or you use some type of editing software, it is essential that you understand basic HTML tags and the procedures for putting your page on the web.

When you save your file, it's critical that you do the following:

- Save your Web file with the extension **.htm** if you are using either Mac or PC as some web browsers do not recognize the .html tag.
- Save your file as **Text Only**. (Note Pad, Simple Text, and UNIX editors automatically save files as Text Only. If you use other types of word processors, you'll have to specify this format under the Save options.) Be sure that the file name ends in .htm and *not* .txt (i.e. home.htm *not* home.txt).

We'll use Note Pad for the PC and Simple Text for the Macintosh in our examples. For this example, begin by saving the file as "MyPage.htm" to a folder (or directory) you've named "HTM" or other name as per your group of lab policy. *Remember that directory name:* you need to keep all files related to your page in the same folder or directory so that file references and links work properly on your page.

Basic Page Structure

Let's look at the tags that are the bare bones of your page:

HTML tags: These tags signal the browser that this is a readable HTML file. These will be the first and final tags in your page source file:

<html>

Your entire document (tags, text and all) goes here.**</html>**

Head tags: These tags identify the header area of your page and enclose the title tags (discussed below):

<head>title</head>

Title tags: The title you specify with these tags appears in the title bar of your Netscape window, and on the page itself in text-only browsers. Make your titles short--up to about six words.

<title>web page title goes here</title>

Body tags: After the title, add the body tags. The body text of your document goes between these two tags, like this:

<body>The body of your page goes between these.</body>

Try using these tags in your new **Mypage.htm** file. Add the following text to the file, substituting your own information for the text in italics :

Exercise 1

```
<HTML>
<Head><Title>Your Name </Title></head>
<body>

</body> </html>
```

Be sure to add the closing (</body> </html>) tags to the very end of your document. This simple bit of code will only generate a blank page with a title in the title bar. The important concept here is that you understand that all tags have a beginning and an end or closing. If you leave out either of these parts or the any part of the tag itself (the angle bracket or the slash), the code will do nothing. Or some *very* strange things depending upon which tag and which part of a tag is missing. Now it's time to begin fleshing out your page.

Formatting Your Page

Add text to your document between the <body></body> tag set. **Keep in mind that none of the spacing or formatting your word-processor uses will be read by a browser.** Each time you hit a tab key or the return key only makes things look nice and neat on your code. In other words, you can type your code without tabs line breaks or paragraph breaks -- your web page code can appear as one continuous block of text with single spacing between words. These key strokes **do not** change the formatting on your web page. Only the HTML formatting tags do. W ith a Web browser, you can't take any formatting for granted, even paragraph breaks and tabs. Let's look at some ways to format your page.

Controlling Text Style

Authoring on the World Wide Web can be like living in a Wild West town. Oh sure, you have a sheriff of sorts, in the form of the folks who come up with HTML standards and try to keep the peace, but there is a certain degree of lawlessness. Perhaps most surprising is that, as a Web author, you don't control the specific font visitors to your page see; individual browsers have that control. In addition, not all browser programs read tags the same way, and some tags only work with a particular browser type. For example, Netscape may read a tag differently than the Mosaic browser, while Lynx reads the tag yet another way.

You do have somewhat limited control of text styles. To control text styles, you can use two basic types of character tags. A "physical" character tag is comparable to styles you're used to using with word-processors; you specify that a block of text is bold or italic, for example. With a "logical" tag, on the other hand, you control the way certain

text looks relative to other text in your document. Logical tags make text stand out, but the browser program and/or the browser user determine(s) exactly how it stands out.

We recommend logical tags for one reason: HTML standards are moving toward logical styles and away from physical styles. New browsing programs will probably be geared toward logical styles, and may eventually ignore physical styles altogether. Here are some ways you can change text styles with logical tags:

Boldface Text: Most, but not all, browsers display text as bold when you use the or tag set, as in this example:

I do ****not**** like green eggs and ham. (logical)
I do ****not**** like green eggs and ham. (physical)

Text in Italics: Many, but not all, browsers display text in italics when you use the or tag sets, as in this example:

My favorite book is ****Green Eggs & Ham****. (logical)
My favorite book is ****Green Eggs & Ham****. (physical)

Use italics sparingly on the Web; italics can be difficult to read on-line.

Headers: Header text has a blank line before and after it, is usually bold, and may be larger than normal body text. Here's an example:

<h1>This is a level one header**</h1>**

You can use header styles 1-6; in relation to normal text, <h1> is the most dramatic, and <h6>, the least.

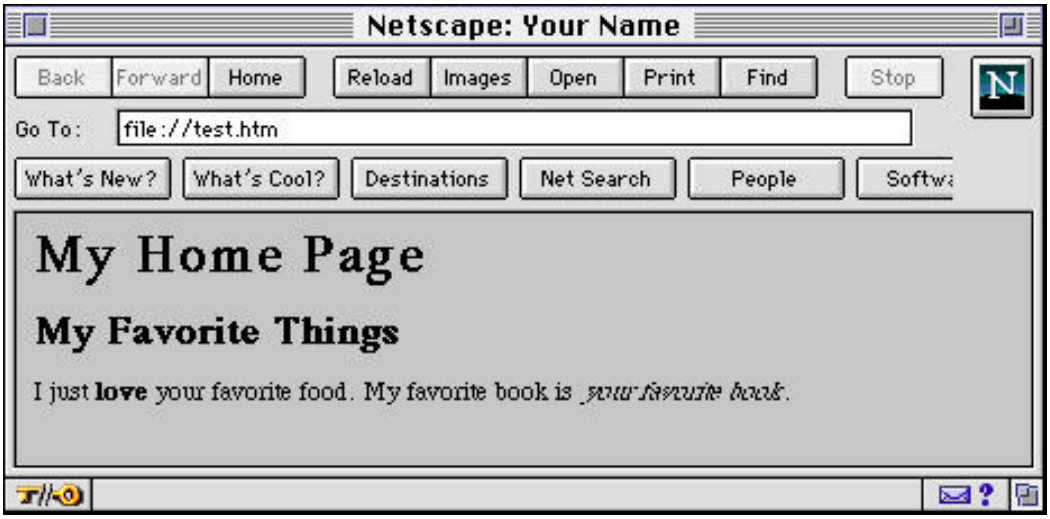
Let's try using these tags on your practice page. Add this text after the <body> tag:

Exercise 2

<h1>My Home Page</h1>
<h2>My Favorite Things</h2>
I just love your favorite food. My favorite book is your favorite book.

Checking Your Work with a Browser

Before you get too far, it's a good idea to check how your file looks with a browser. First, save your document again, and then start Netscape. From the Netscape **File** menu, select **Open file ...** and open your file (MyPage.htm in the folder HTM). Your page should look something like this:



If you have trouble, make sure that the file is **Text Only**, and that you actually saved the file. You ***must*** remember to save after each change you make in the word processor. Also, try clicking on the **Reload** button in Netscape. Each time you make a change or addition to your page, save the change, switch back to the Netscape window, and click on the **Reload** button to view the new version of your file. Get in the habit of checking your page as you add to it.

Controlling Spacing

Use the following tags to control paragraph and line spacing on your page:

Line Break: To start a new line of text, or to break a line at a specific point, use this tag:

**
**

Note: A line break is a single tag and has no closing or end tag.

Paragraph: Use a paragraph tag to separate one paragraph from another. first paragraph of text **<p>**next paragraph A paragraph tag adds a blank line within a block of text, usually to distinguish one paragraph from another. Remember: browsers ignore blank lines (the return key) or space bar or tabs in the source text.

Block Quotes: Use the **<blockquote></blockquote>** tags to set paragraphs apart from other text. This tag set indents a block of text from the left and right margins, and inserts a single line space above and below the block. Be sure to add the closing tag, or you'll indent more than you intend to.

Let's try adding some text to your practice document. Return to Myfile.htm, and after the tag, add the following information to your page:

Exercise 3

<p>My favorite way to spend a Saturday night is fill in the blank.
<h2>Music</h2>

<blockquote>I prefer your favorite type of music here and another favorite here. I can't stand your least favorite music here.
</blockquote>

One More Way to Control Spacing: Pre-formatted Text

Use this tag set to preserve the tabs, indentations, and line spacing in your original source text.

<pre>Preformatted Text goes here</pre>

Graphical browsers like Netscape display pre-formatted text in a fixed font, such as Courier. The browser user controls which specific font displays by selecting an option in the Netscape **Preferences** menu.

Try adding this text to your document, and see how it looks on Netscape:

Exercise 4

<h2>Pre-Formatting Text</h2>
<pre>I will start a new line of text right here and it will stay when I look at my page in Netscape. I can put a tab right here, and it stays put, too.
</pre>

Ordered and Unordered Lists

Web authors frequently use these two basic list types on their pages:

1. **Unordered lists are bulleted**. Use unordered lists when you want to list parts of a whole, but don't want to imply a sequence of steps or a specific order.
2. **Ordered lists are numbered**. Use ordered lists when the list has an order or sequence; for example, a list of steps or a "Top 10" list.

To create a list, you use two types of tags. The first set of tags defines the list itself. The next set of tags defines the items in the list.

Adding an Unordered List

To add an unordered (bulleted) list to your page, you first define the list itself with **
**, and then specify the items in the list with <li>, as shown here:

```
<ul>
<li> List item
<li> List item
</ul>
```

Notice that we used a closing tag for the list itself, but not for the list items. In this example, we also put each part of our list on a separate line. This helps us keep track of the list in the source file, but the browser ignores these line breaks. We could have put all this information on one line. Remember, only the markup tags determine formatting, though human beings tend to prefer things to be visually ordered for ease of reading.

Now try adding the following unordered list to your practice Web page and review your work:

Exercise 5

```
<h2>Things I Do</h2>
I stay busy at my job.  Things I do in a week include:
<ul>
<li>Meetings
<li>Adminstrative duties
<li>Primary duties
</ul>
```

Adding an ordered list

Set up an ordered list the same way as an unordered list, with one exception: in an ordered list, the list tag set is **
**. Here's an example:

```
<ol> <li> List item1 <li> List item2 <li> List item3 </ol>
```

Even though we didn't separate each item this time, the browser displays each item on a separate line. Try adding the following list to your practice page:

Exercise 6

```
<h2>Restaurants</h2>
My 3 favorite places to eat in RTP, in order, are:
<ol>
<li>Favorite #1
<li>Favorite #2
<li>Favorite #3 </ol>
```

A Third List Type: Definition Lists

Definition lists are intended for definitions, but you can use them anytime you want to put a paragraph with a heading on your page. You begin the definition list itself with the **<dl>** tag. Then add the **<dt>** tag and the term to be defined, or any heading text. Next add the **<dd>** tag and the text of the definition or paragraph. Finally, add the **</dl>** end tag. Here's an example of a defintion list:

Exercise 7

```
<dl>
<dt>My Life:  The Early Years
<dd>I think Bill Cosby said it best:  I started my life as a child.
</dl>
```

Try adding a definition list to your page on your own, and take a look at how it looks with Netscape.

Adding a Horizontal Line or Rule

If you want to visually separate text, you can add a horizontal line anywhere in your document with the **<hr>** tag .

Let's try it in your practice document. Add this tag to the end of Mypage.htm, just before the </body> tag:

Exercise 8

```
<hr>
```

You can use sizes 1-6; in relation to normal text, **<hr size=1>** is the most dramatic, and **<hr size=6>**, the least.

You may also choose the width (the percentage of the window covered by the line from left to right). An example of this tag would be:

<hr width="50%">.

This tag requires the addition of quotation marks and the percentage sign to understand what size you would like. Without either of these two portions of this tag, your web page may act strangely.

Review your work just as you have previously.

Providing Text Links to Other Pages and Files

One of the great features of the World Wide Web is its inter-connectivity. You can provide links to other pages around the world by using this basic tag format:

words/link the user sees

The link (link the user sees) is also called the "anchor." In this document, our link examples will use anchor text; in the next chapter, you'll learn how to make graphics into links. This tag requires the addition of quotation marks to understand where it is to find the referenced link. Without this portion of the tag, your web page may act strangely.

Providing Text Links to Other Pages

To provide a text link to another Web page anywhere in the world, use this format:

anchor text

where full URL address is the URL of the page the browser displays when the user clicks on the anchor text, (that is, the link).

Note: URL addresses are case sensitive, so be careful typing the URL. When possible, visit the page to which you're linking, copy the URL from the Location field, and paste the URL into your document.) Give your links meaningful names, so users can quickly scan your page to find the links they want.

Try adding this link to your practice page:

Exercise 9

<h2>Favorite Links </h2>

One of my favorite links is the The Weather Channel.

View your work as before. Click on the anchor text to verify that the URL is correct and your link takes you where you want to go.

Adding an In-line Image

An "in-line" image is a graphics file (either a .gif or .jpg image file) that displays as part of your HTML document. It appears automatically if the user turns on "Auto-Load images" in the **Options** menu. If that option isn't turned on, a place holder appears instead, and the graphics file can be downloaded later. To add an in-line image, you specify the image source using this format:

Exercise 10

<img src="filename.gif">

Note: Replace *filename.gif* with the real name of your graphics file. **Be sure** to include the quotation marks around the graphic file name and the subdirectory path if you keep your graphics file inside an images folder (ex. <img src="/images/filename.gif">).

Let's try adding a graphic to your page. At the end of your practice file, add the following:

Exercise 11
<img src="star.gif"><p>

View your file as before (remember to save the file and reload it in Netscape.) If "Auto-Load images" is turned on, you'll see the graphic itself; if it's turned off, you'll see a place holder, which you can click to view the graphic.

Note: You must have "practice.gif" in the same directory as the HTML document for this link to work.

Providing Links to Different Sections of Your Page

If your Web page is long, you'll need to break it into logical sections, and provide links that make it easy to move from section to section. To see what we mean, take a look at the NIEHS Computer page at

http://www.niehs.nih.gov/websmith/webtools/imagview.htm

Scroll down to the Table of Contents. Each item in the table is a link to a major category within the document.

Let's look at how to provide links within a document, using a table of contents as an example. First, you create the link to your Table of Contents entry. Next, you create the target of the link somewhere in your document. The format for a link to an internal section of your document is very similar to other links:

**anchor text user sees **

"Name" is the the target of your link, which we specify in the second step. Note: "#" is crucial here.

We'll use the following format to add the name somewhere else in the document:

*Text user sees *

The "name" is the one we use in the anchor reference, but this time, we don't type "#" before it. Now when you follow the link, the browser displays this section of your page. Notice that, unlike the table of contents reference text, your "named" text does not appear as a hyperlink.

Finishing Touches

Signing Your Work

Signing your work tells visitors to your page who is responsible for its content. Web authors often sign their pages with an e-mail address, which enables people to contact them with questions or suggestions about their pages. You can either provide only your e-mail address, or your e-mail address with a link to a form readers can use to send mail directly from their browser to your e-mail account. Try adding an e-mail link to your page:

your_email@niehs.nih.gov

Readers can follow the link to a form that sends e-mail to the address specified in the tag. This form works with most browsers, including Netscape and Lynx (if this is to be an external page). Try adding an e-mail link to your practice document.

Completing Your Page

Because the Web changes constantly, it's a good idea to let your readers know how current your information is. Include the date that you last revised the source document at the end of your page. See the **HTML Guidelines** page for the suggested date format.

http://www.niehs.nih.gov/websmith/restrict/policy.htm

http://www.niehs.nih.gov/websmith/examples/example.htm

*Revised on: date
*

Summary of Tags Discussed in This Chapter:

Tag Pair or Group	Purpose	Notes
<html>page</html>	Identifies page as html.	Required.
<head>title tags</head>	Identifies heading area.	Required
<title>title of page</title>	Title of page. Appears in title bar.	Required
<body>body of page</body>	Body text of page.	Required
text	Heading text.	Can use h1-h6.
text	Boldface (most browsers)	
text	Italic (most browsers)	
 	Line Break	
<p>	Paragraph Break	
<blockquote>text</blockquote>	Indents text and inserts line space before and after tagged section.	
<pre>text</pre>	Retains paragraph formatting of original file.	Fixed font.
	Unordered list (bulleted).	
/ol>	Ordered list (numbered).	
text	List item (for and tags).	
<dl></dl>	Definition list.	
<dt>text	Definition term (for <dl> tag).	
<dd>text	Definition (for <dl> tag).	
<hr> <hr size=1> <hr width="50%">	Horizontal line (rule).	Can use sizes 1-6 and width tags.
	Link to location enclosed by quotation marks.	
<img src="file_location">	Display image.	
text	Link to location on current page or graphic.	
text or graphic	Target of link to location on page.	

Chapter 2
Putting Your Page on the Web

Now that you've had a chance to practice making some basic web pages, you'll probably want to upload the pages and make them "live." When you are working on your web pages from a networked NIEHS computer, the easiest way to upload them is to take advantage of **Chooser** (Mac) or **Network Neighborhood** (PC) which connects you directly to either of the two web servers.

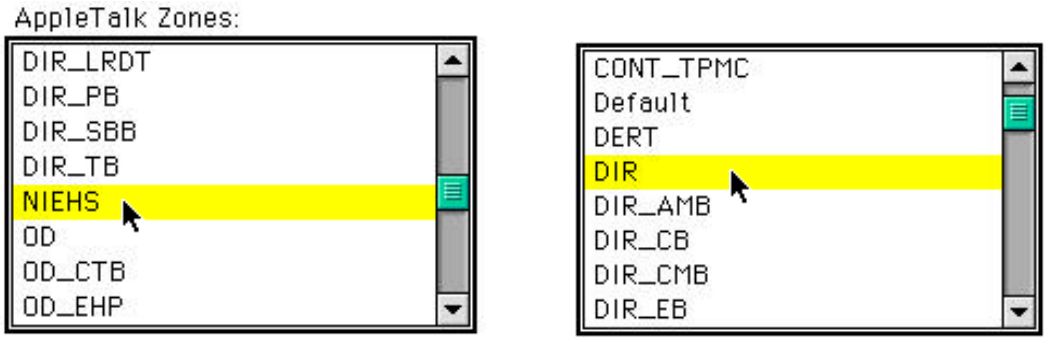
Steps for Uploading Web Page Files on a Macintosh

1. Create and save the files.

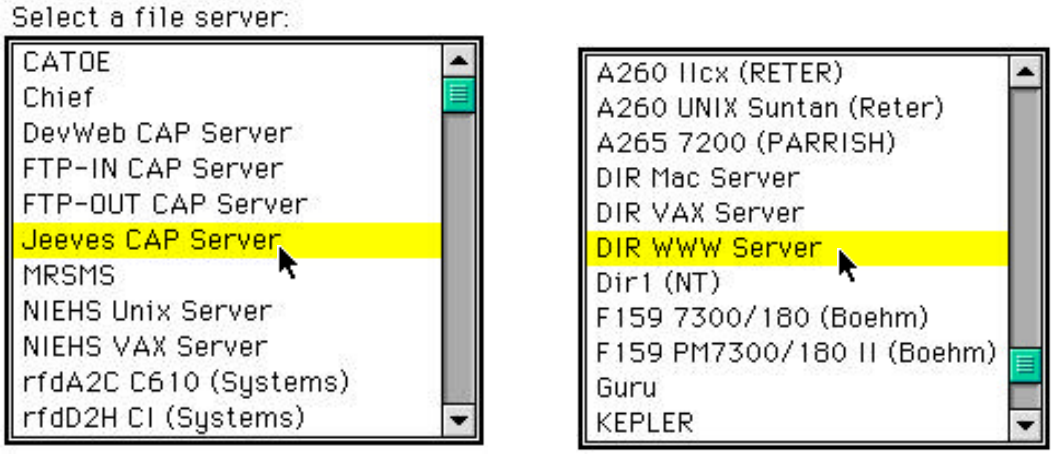
2. From the Apple menu items  , select the **Chooser** 

3. In the top left-hand side of the window that opens, click on the **AppleShare** icon:

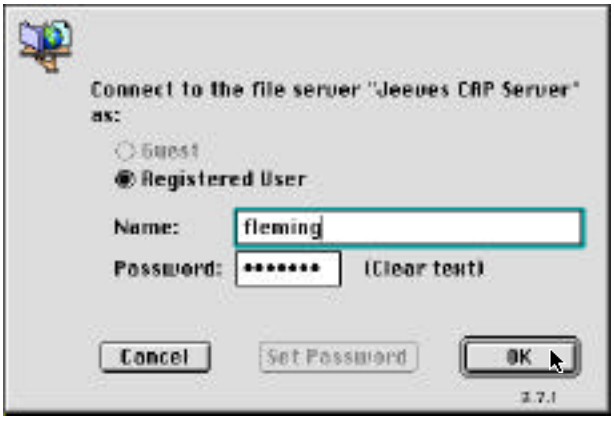
4. In the **AppleTalk Zones** section of the sreen, select **NIEHS** if you have a Jeeves account OR select **DIR** if you have a DIR web page account.



5. From the **Select a File Server** section, choose the **Jeeves CAP Server** or the **DIR WWW Server**, as appropriate.


Select a file server:
CATOE
Chief
DevWeb CAP Server
FTP-IN CAP Server
FTP-OUT CAP Server
Jeeves CAP Server
MRSMS
NIEHS Unix Server
NIEHS VAX Server
rfdA2C C610 (Systems)
rfdD2H CI (Systems)
A260 IIcx (RETER)
A260 UNIX Suntan (Reter)
A265 7200 (PARRISH)
DIR Mac Server
DIR VAX Server
DIR WWW Server
Dir1 (NT)
F159 7300/180 (Boehm)
F159 PM7300/180 II (Boehm)
Guru
KEPLER

6. Type in your **Jeeves** or your **DIR** User ID and Password and the click on **OK**.

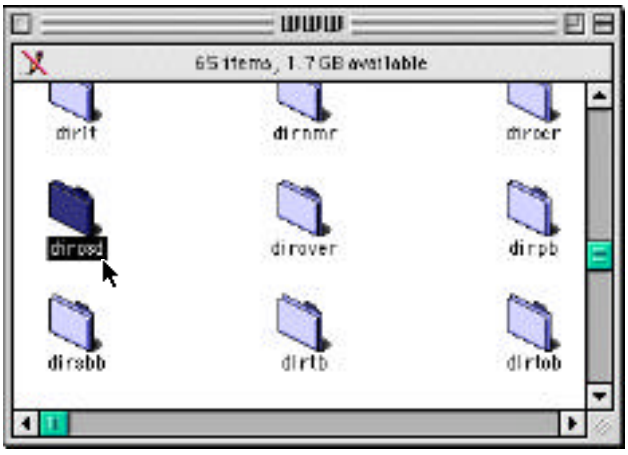


7. When the next window opens, select **WWW** and click on **OK**.

Note: Do not put the check mark in the box to the right unless you want your computer to open up the Jeeves server every time you start up your machine.

8. Close Chooser and locate the **WWW server icon** on your desktop and double-click.

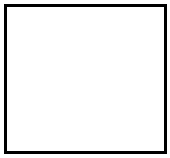
9. Locate the folder you have been given permission to access for your group's web pages and copy your web files to this location. To do this, you can simply drag your files (.htm, .jpg and .gif) into your folder on the server to upload them or "make them live."



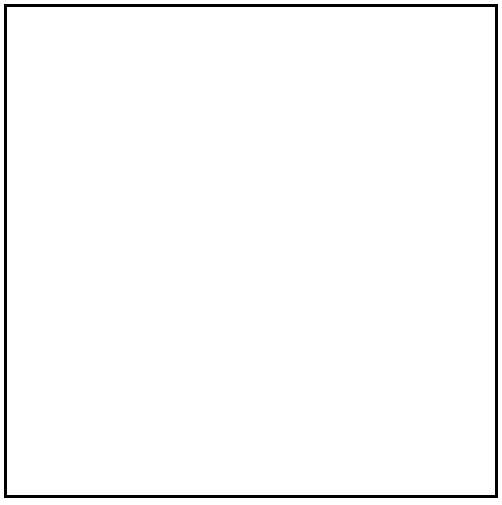
Steps for Uploading Web Page Files on a PC

1. Create and save the files in appropriate format and with appropriate extsion.

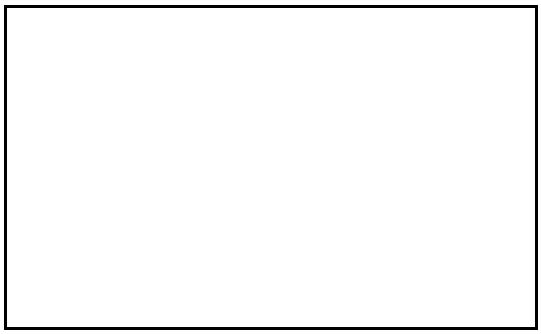
2. On your desktop, double-click on the **Network Neighborhood** icon:

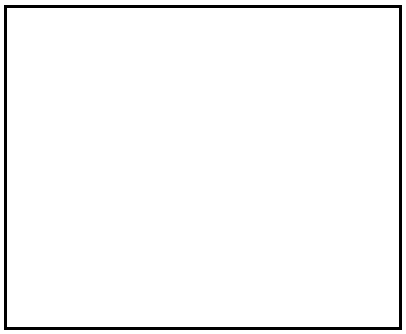


3. When the window opens, select **Entire Network**.

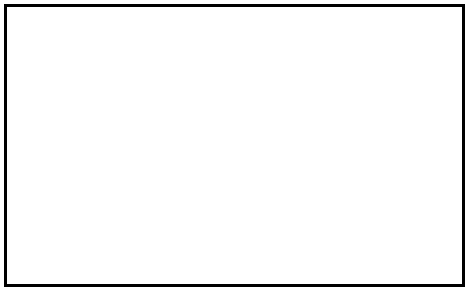


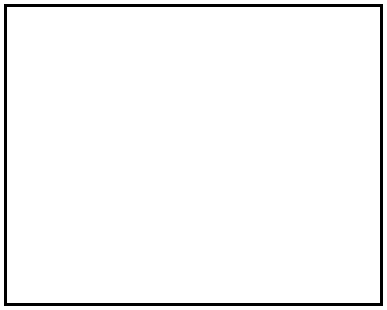
4. When the next window opens, scroll over and select **NIEHS**.for Jeeves or **DIR**.



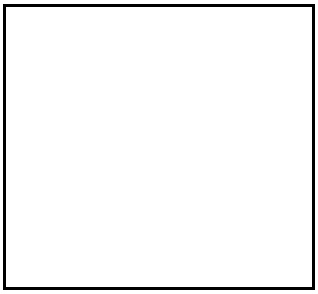


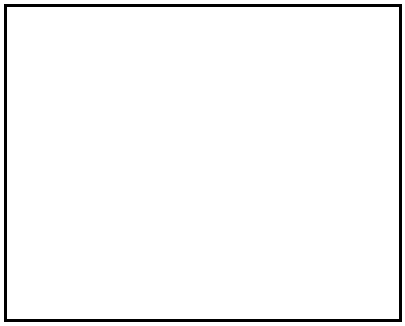
5. When the next window opens, scroll over and select **Jeeves** or **Bedlam**.



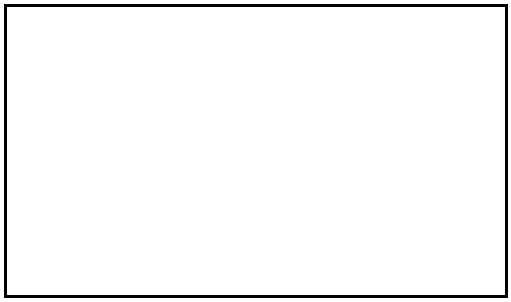


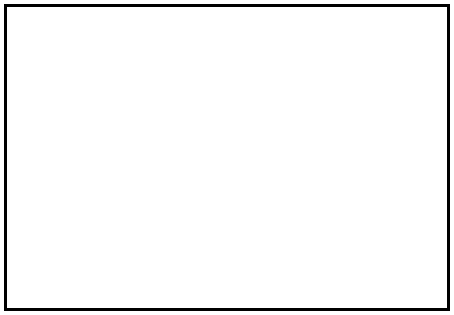
5. When the next window opens, scroll over and select **WWW**.





6. Open the folder for which you've been given permission to work with.

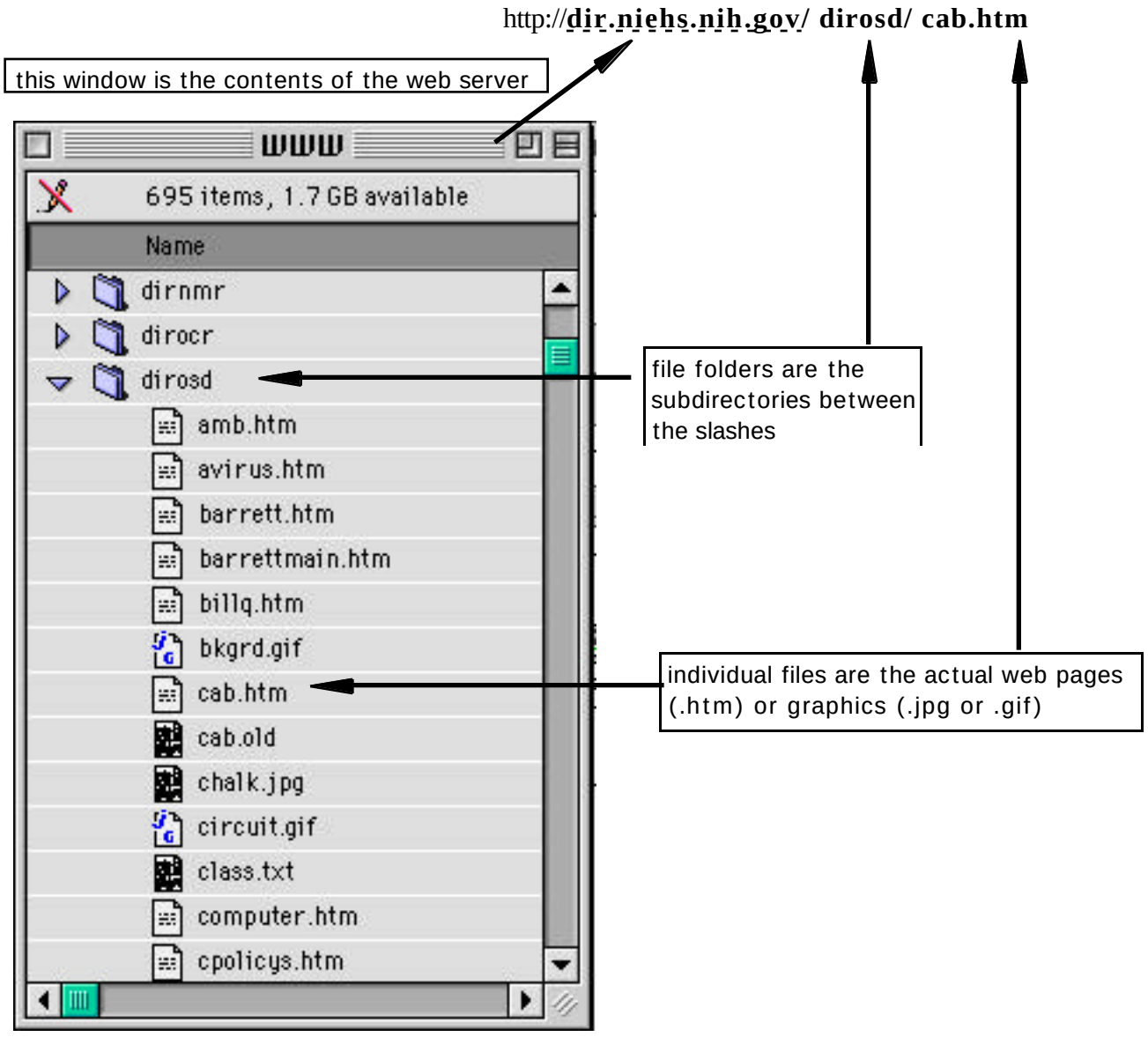




7. Locate the folder you have been given permission to access for your group's web pages and copy your web files to this location. To do this, you can simply drag your files (.htm, .jpg and .gif) into your folder on the server to upload them or "make them live."

Basic Structure

Once you are in your directory, you will wan to have a basic layout that makes it easy for you to locate your files in the future. It is a good idea to have a directory to store all your image files. A basic file structure may look something like this:



The remainder of this chapter is designed to teach you how to make your pages “live” so that other people around the world can see them using other software programs. This is especially handy if you have a university or commercial web page account..

There are two hard and fast rules that **must** be followed in order to properly upload your page and set the permissions.

1. Upload your web page and image files to the web server (Jeeves)
2. Set the permissions on each of these files

Because there are a variety of software packages used for uploading and setting permissions, these actions will be discussed in general terms. There are site licensed, built-in applications, and shareware applications that can be used for these two step of web page authoring.

Rules of the game:

Upload each of these files to the appropriate directory space for your web pages (using an FTP client such as **Fetch** for the Macintosh and **WSFTP** for Windows users).

- All web pages must be saved as and uploaded as **text** files, but have the **.htm** or **.html** extension. (Remember to use eight character file names if at all possible.)

- All graphics must be uploaded as **raw data** and have the appropriate **.jpg** or **.gif** extension.

Set permissions on the files. (many FTP clients have this function built in, otherwise use a Telnet client)

- Permission to read should be **all**.

- Permission to write should be either **owner** or **owner and group**.

- Permission to execute should be either **owner** or **owner and group**.

Terms You Need to Know

The following table is for your reference as you learn about telnet and FTP:

Term	*Definition*
Client	A program you use to request a resource or service from another program called a "server." For example, you may use a client to request a file from a file server. Generally, the client is located on a different computer from the server, often on your own desktop. See "Server."
Download	Simply put, to "get" a file from a server. To be more specific, it means to transfer files between two computers, usually, to transfer files from a server on a larger computer to a client on a smaller computer. See "Upload."
Host	A computer, connected to a network, that allows you to log in and use resources. For example, Isis is a host computer you may use for your email service.
IP Number	**Internet Protocol number**. The unique number that identifies a computer on the Internet. You can use a computer's name or its IP number when you specify its address.
Server	A program that "serves up" resources and services you request using your client program. The server is usually located on a "host" computer. "Host" and "Server" are frequently used interchangeably. See "Client."
Upload	Simply put, to "put" a file onto a server. To be more specific, it means to transfer files between two computers, usually, to transfer files from a client on a smaller computer to a server on a larger computer. See "Download."

We got help with these definitions from the Free On-line Dictionary of Computing at:

http://wombat.doc.ic.ac.uk/

Just what are FTP andTelnet?

Telnet and FTP are protocols, a set of formal rules for transmitting data. You use these protocols to connect to servers on the Internet for two different purposes:

- To log on to an Internet server and look at information there, you use the telnet protocol. For example, you use Telnet to log on to the web server Jeeves. You cannot download or upload files using Telnet.
- To download or upload files, you use the File Transfer Protocol (FTP).

FTP

You can use FTP to transfer files between:

- Your computer and a server on which you have an account
- Your accounts on two different servers
- Your computer (or account) and an "anonymous FTP" server

Anonymous FTP servers allow you to log on and transfer files, even though you don't have an account on the servers. Let's look at some reasons you might use FTP.

If you have an account on a server on campus, you can use FTP to upload files from your PC to your account. With the file in your account, you might then want to email it, for example. Downloading files through anonymous FTP is a good way to get free software and other types of files. (Note: before you run any software you download, be sure to check it for viruses. When you download text files, however, you don't need to worry about infecting your computer with virus; viruses are transmitted only through program files. Some servers also allow the public to upload files to selected directories--which is why you need to be careful about viruses when you download programs.)

Using Fetch

First Things First: Connect to a Host

To start Fetch, double-click on the Fetch icon.

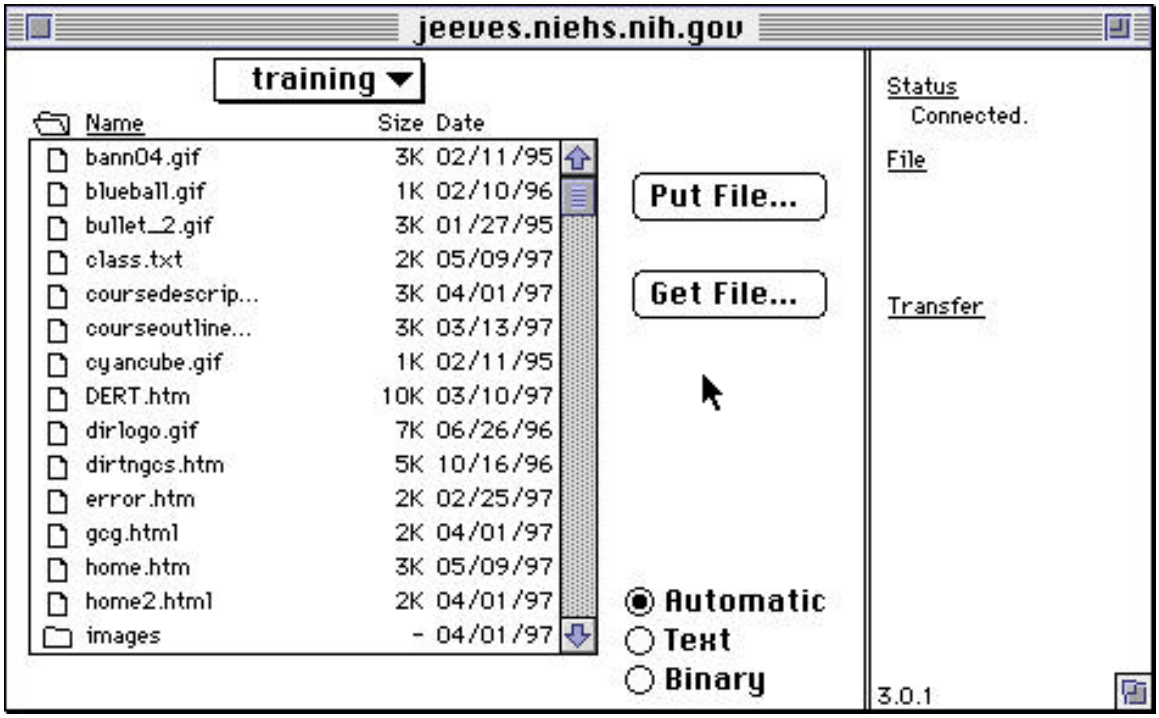
This dialog box pops up:

Use the dialog box to log on to the host you choose. Type the name (jeeves.niehs.nih.gov) or IP number of the host in the Host field. Next, complete the User ID and Password fields in one of two ways:

To go immediately to the directory you want, you can complete the Directory field, but you must type each letter in the correct case (lower = abc, upper = ABC). The directory on Jeeves for web pages begins as follows: usr/local/WWW/html/*yourgroupname*. It may be easier to move through the directories after you make your connection. When you're ready, click on **OK**.

Downloading and Uploading Files

When Fetch connects to the host, you'll see a window like this one:

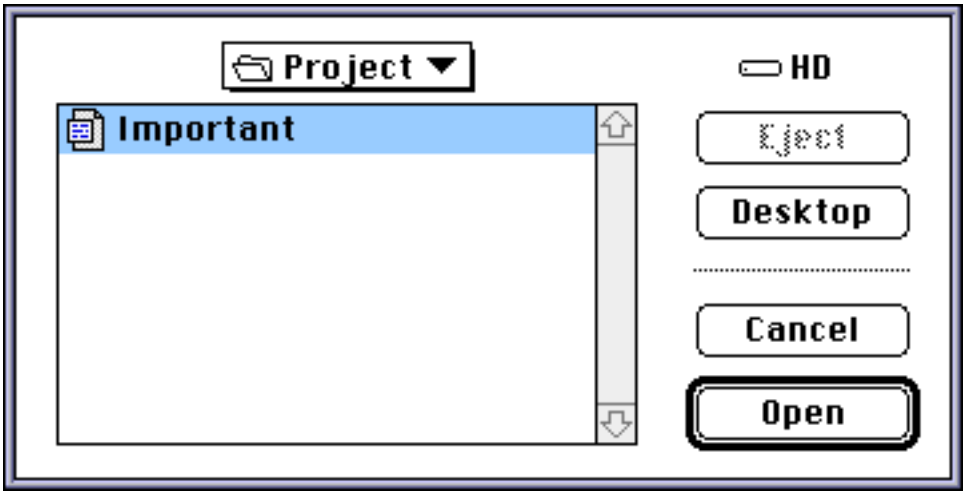


You can move around in the server's files by changing the current directory (click on the box displayed above the list of files), and by double-clicking on the different folders to display their contents. Notice that you can either "**Get**" (download) a file from a server, or "**Put**" (upload) a file on a server.

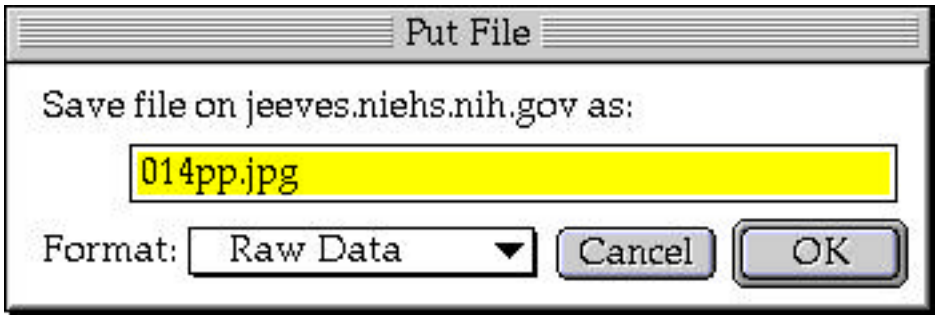
Steps for Uploading on a Macintosh

The process for uploading files with Fetch is similar to downloading:

1. Connect to a host (jeeves.niehs.nih.gov). Using your user ID and password
2. Locate the directory on the server where you want to put your file. (It is going to begin with usr/local/WWW/html/*yourgroup* for web pags.)
3. Click on **Put**. A window pops up showing files available to you locally:



4. Select the file, files, or directory you want to put onto the server (upload). Then click on **Open**. This window pops up:



5. Now you can save a copy of the file to the directory you chose on the host. You may want to change the file name or format. In the example shown, we're uploading a regular graphic file used in a web page.

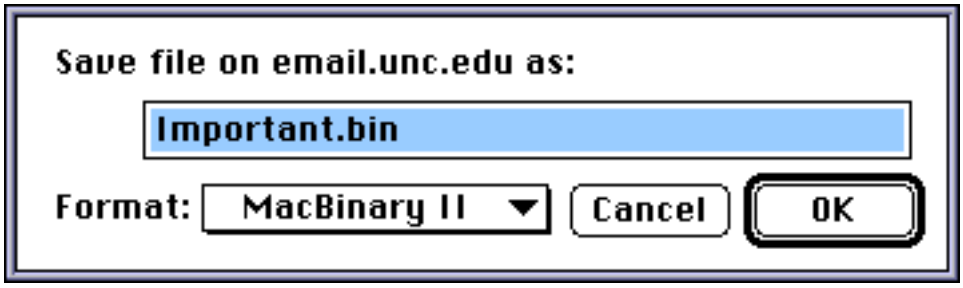
   Note: You **must** to change the format to Text by pulling down the arrow and selecting **Text** instead of the default MacBinary II for .htm files and **Raw Data** for .jpg or .gif files.

When you click on **OK**, Fetch completes the uploading process.

Steps for Downloading on a Macintosh

To download, do the following:

1. Select the file, group of files, or directory you want.
2. Specify the file type by selecting Automatic, Text, or Binary from the **Format** pull-down.
   - For **.htm** files, select **Text**.
   - For **graphics** files, select **Raw Data**.
3. Click on **Get**. A window pops up that allows you to save the file.



4.Select the file's destination, and click on **Save**.

Fetch Menu Commands

The following table below lists some ways you can customize your Fetch session.

If you want to . . .	then choose and select . . .	this menu . . .
See file sizes, the date files were last saved, and other detailed file information	Remote	View File List
Preview the contents of a file; select the file and . . .	Remote	View File
Select files from different folders to download	Remote	Put Folders and Files . . .
Specify a Host connection that comes up automatically when you start Fetch (default connection)	Customize	Edit Shortcuts

For more options, refer to Fetch Help.

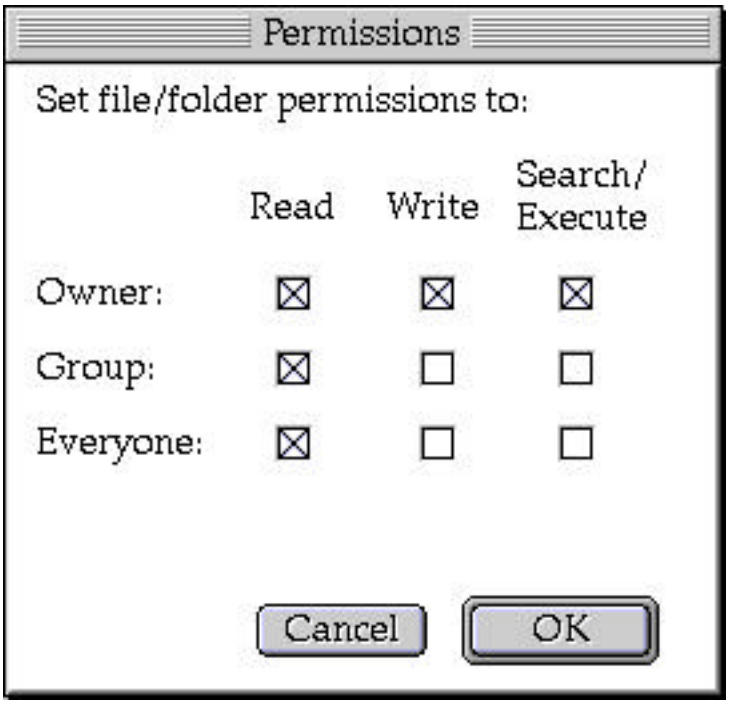
Closing Your Connection

Click on **Close Connection** in the Fetch main window to close your connection to a host. If you're visiting an anonymous FTP server, it's especially important that you close your connection as soon as you've transferred the files you need. FTP servers limit the number of connections that can be made at any one time, so you open up a space for someone else when you close your connection. When you've completed all your sessions, you can quit Fetch by selecting Quit from the File menu.

You can also transfer files on the World Wide Web using FTP. Details of the FTP session vary with the browser you use, but generally, you simply point and click to select a file to download. Of course, the file must be available at an FTP server for you to be able to download it. The browser steps you through choices for downloading, depending on your computer setup.

Making Files Usable

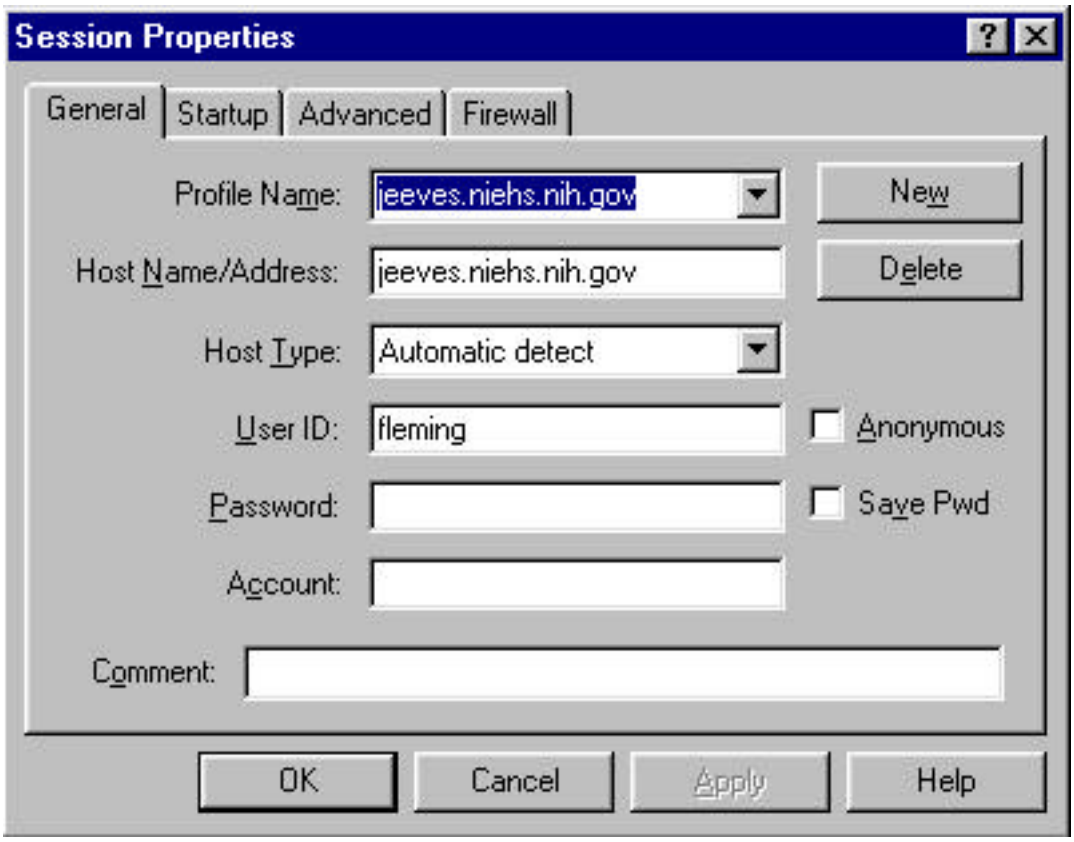
Once you've uploaded your web pages and associated graphics, you must set permissions. Under most FTP/Fetch programs, there is a menu item such as **Remote** where you can select **Set Permissions**.



Always give the "owner" and your "group" to read, write and use the file. Give all others read only permission.

Steps for Uploading and Downloading on a PC

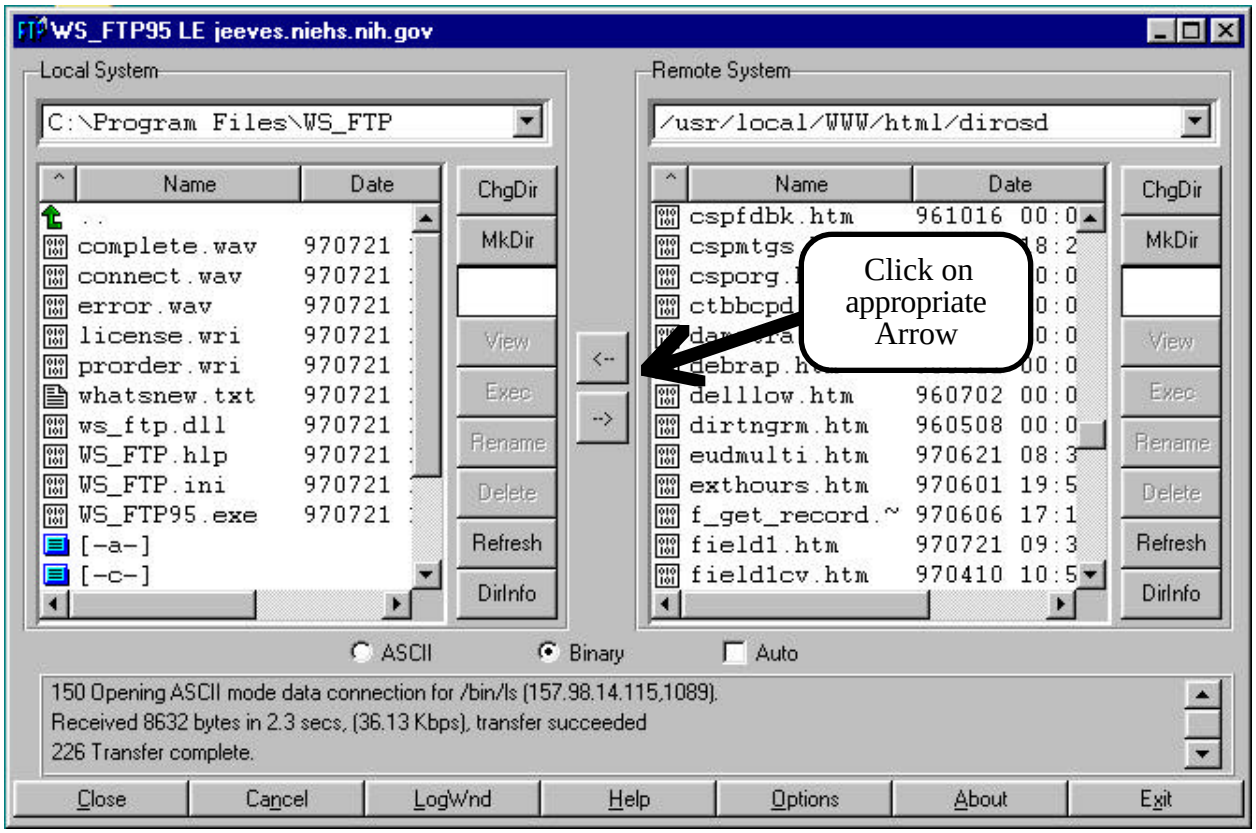
1. Connect to a host (jeeves.niehs.nih.gov). Using your user ID and password.
2. Locate the directory on the server where you want to put your file. (It is going to begin with usr/local/WWW/html/*yourgroup* for web pags.)



3. Click on **upload arrow** (the one that is pointing from left hand window displaying files on your computer to the Jeeves list on the right-hand window).

   OR - Click on **download arrow** (the one that is pointing to left hand window displaying files on your computer from the Jeeves list on the right-hand window).

A window pops up showing files available to you locally:



File Names

File names in the UNIX operating system can be much longer than file names in Windows. For example, you can name a file "Myhomepage.htm" in UNIX, but not in Windows 3.x. If you're downloading a file that doesn't follow Windows naming conventions, rename it. If you have appropriate access privileges, you can also rename a file in a directory by clicking on it, selecting the Rename button, and completing the Rename Files window that pops up.

Drop and Drag

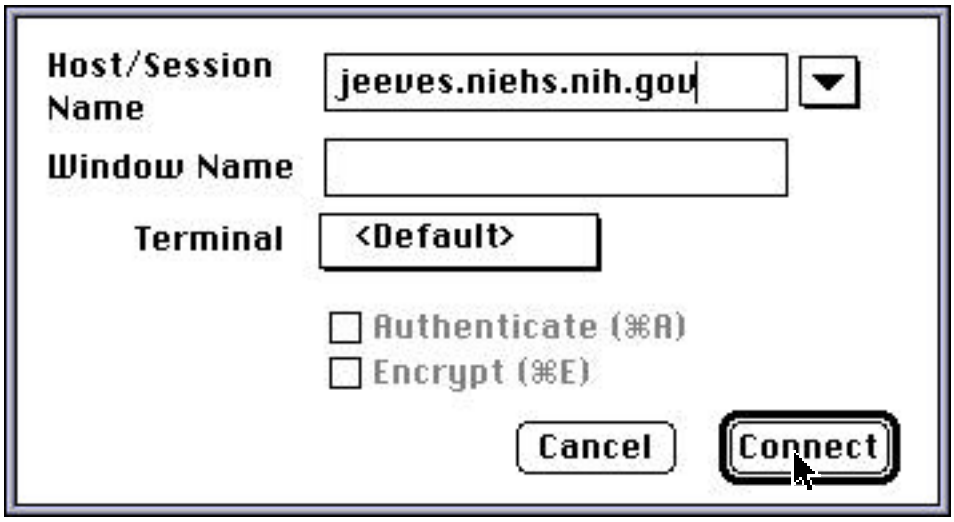
You can also find the Jeeves server under the Chooser on Macintoshes and through the Network Neighborhood on PC. When you have logged into Jeeves through this method, you may simply drop and drag your web pages and associated graphics.

Telnet

Making a Telnet Connection With NCSA Telnet

Macintosh
To telnet to a host from a Macintosh with a direct Internet connection, you can use a program called NCSA Telnet (ask your CSP for assistance if you do not have this program, or find it at http://www.shareware.com).

To see how NCSA Telnet works, try telnetting to the library server. Double-click on the NCSA Telnet icon to start the program. This dialog box pops up:



In the Host/Session Name field, type the name or IP number of the host you want to access. For example, to access the jeeves server, type **jeeves.niehs.nih.gov**. Then click on the **Connect** button.

PC/Windows 95
Telnet is built-in to Windows 95 and can be accessed by going to the **Start menu**, choosing **Programs**, then selecting **MS-DOS**. When the DOS prompt (C>) appears, type in:

telnet jeeves.niehs.nih.gov

Moving Around and Exiting

Once you have signed using your login ID and your password, you will be in the wild and wonderful world of UNIX. This operating system has specific commands that will enable you to see your files and set permissions similar to DOS (the operating system on Pcs that came before any version of Windows). Be warned that if you do know DOS, that you should become familiar with UNIX commands and not assume that they are the same!

Telnet's Limitation

You can get access to a great deal of information and services through telnet, but you cannot transfer files. To transfer files, use the File Transfer Protocol (FTP), which is discussed next.

USEFUL UNIX COMMANDS FOR WORKING WITH WEB PAGES:

Summary of Basic UNIX Commands

Below is a reference list of the UNIX commands you will use most often and that are necessary for a basic use of UNIX. You may wish to print out a hard copy of this section and keep it in a handy place near your computer. Keep in mind that command syntax is:

command -flag(s) argument(s)

ls (list) Provides a directory listing.

ls	lists a directory in columns
ls -l	gives a fuller listing including file permisssions, size, date created
ls -al	similar to the above but includes "dot"/hidden files

cd (change directory)

cd	returns you to the home directory
cd ..	moves up one directory level
cd ../.	moves up two directory levels
cd *subdirectory*	moves to named subdirectory

cp (copy)

cp *file1 file2*	copies file1 to file 2
cp *file1 directory*	copies file 1 to named directory
cp *file1 file2 directory*	copies file1 and file2 to named directory

mv (move) Moves or renames one or more files.

mv *file1 newname*	renames file1
mv *directory newname*	renames directory
mv *file1 directory*	moves file to named directory
mv *file1 directory/newname*	moves file1 to named directory and renames

rm (remove) Deletes files*

rm *filename*	deletes the named file
rm -i *filename*	deletes named file after prompting to make sure you wish to remove it

*BE EXTREMELY CAREFUL with **rm** and never use **rm***.
There is no undelete command!

mkdir (make directory)

mkdir *subdirectory* creates a directory within the current directory

rmdir (remove directory)

rmdir *subdirectory* deletes directory within the current directory (if directory is empty)

man (manual) Displays help or manual documentation.

man *command* displays manual information for the specified command
man -k *keyword* displays 1 line synopsis of each manual section referring to the keyword

more Lists files one screen at a time.

more *filename* scrolls forward through file one screen at a time. Press spacebar to go forward one screen or q to quit.

pwd (print working directory) Displays path name of current directory.

pwd displays your current directory Create/Delete File

Create/Delete Directory

To create a new directory, just use the mkdir command with the new directory's name. To create "my_new_dir", type mkdir my_new_dir. To remove the directory you just created, type rmdir my_new_dir. (Directories must be empty before you can delete them with the rmdir command.)

Move/Copy A File

Use the mv command to move a file from one directory to another. The mv command takes two arguments; the name of the file you want to move followed by a destination directory. For example, to move "old_file" into the "archive" directory, type: **mv old_file archive**. (Note: if the directory archive did not exist, the mv command merely renames your file, keeping it in the current directory). When you move a file into a new directory, it retains the name it had in the original directory.

The **cp** (that's UNIX-speak for copy) command works similarly to the **mv** command, however, it retains the original version.

Exit a Telnet Session

exit typed at the command line followed by pressing the return or enter key will end the Telnet session.

Permissions

New users on a UNIX system are often unaware of file security. Without knowing it, they leave sensitive information unprotected. Other users on their machine (or from the rest of the world) access and sometimes even change their files. To combat this problem, consider

the following advice. It is useful to know how secure your files are. To find out, use the **ls -al** command. The security information is contained in the collection of **d**'s, **r**'s, **w**'s, and **x**'s to the left of the output. These ten spaces, containing either letters or dashes, tell you what kind of file is shown and how secure it is. Every file and directory under Unix has a set of permissions associated with it that is shown as a three digit number (such as 755) or combinations of letters (such as a+r). These permissions are categorized into three groups who have or do not have the permissions:

- the file **owner**
- the owner's **group**
- **everyone else** (referred to as "other," "all," or "the world")

These three groups, in turn, may or may not have three different priviledges:

- **privilege to read**--that is, others can only read your files; they cannot alter them
- **privilege to write**--in Unix, this privilege means that others can change, save, or delete your files and directories
- **privilege to execute**--this permission is necessary for other users to be able to search for files within a directory; thus if you do not provide execute privileges for a directory, other users cannot gain access to the files and subdirectories located therein. The execute privilege is also necessary for other users to be able to run files as programs.

When you type the **ls -l** unix command in you will see a string of file namesAn example from a group of web files here at NIEHS:

-rwxrwxrwx	1 reter	www-osd	2662	Aug 19 1996	teal_paper.gif
-rwxr--r--	1 fleming	www-osd	828	Mar 10 14:24	thanks.htm
-rwxrwxrwx	1 reter	www-osd	1204	Jan 31 11:17	thankyou.htm
-rwxr--r--	1 fleming	www-osd	2047	Apr 22 10:47	videos.html
-rwxrw-r--	1 reter	www-osd	6596	Feb 11 1996	w95intro.htm
-rwxrw-r--	1 reter	www-osd	7002	Feb 11 1996	w95net.htm

Let's start with the first column. If you see a '**d**' here, it means the object you are looking at is a subdirectory. If you see a '**-**', the object is a file of some sort (either program or data). The next nine letters detail how secure your files are. The first group of three letters describes *your own privileges.* An '**r**' means you can read a file, a '**w**' means you can write to a file (write means you can edit and delete it), an '**x**' means you can run the file (if it is a program -- an '**x**' setting for a data file will only cause problems), and a '**-**' means that you are restricted from using a file. Almost always, you want to have '**rw-**' for your data files, and '**rwx**' for your programs.

The next set of three letters describes what *members of your group* can do. Groups are special constructs which allow members of a team to share files with their own set of access restrictions. The WebSmith and the SysAdmin have placed you in the correct group and you should never attempt to change the group you are in on your own. Finally, the last three letters are by far the most important, as they represent what *anyone on the system* can do to your files. By default, many machines set permissions like those of 'myfile1' shown above -- everyone can read the file, even though they cannot edit / delete it or run it as a program. For sensitive information, this is obviously undesirable. You can change any of

these permission with the **chmod** command. The command has three parts: whose permissions you want to change, what you want to change them to, and which files to change. Here are some examples:

Doing the Math

If you prefer to set up your permissions using the numeric option, you will need to follow the see the breakdown of how the “-rwxrw-r--” are produced.

Thus from the instructions above, you will be ignoring the first hypen or letter so that there are only nine total variables that concern you:

- permission for the owner to read the file, write to the file, and execute the file;
- permission for the owner's group to read the file, write to the file, and execute the file;
- and permission for others to read the file, write to the file, and execute the file.

These variables are organized into a three by three array as follows:

	owner	group	other
read	4	4	4
write	2	2	2
execute	1	1	1
total "value":	7	7	7

Column Values

The three by three array above shows the basis for describing the set of nine permissions. Note that each permission has a numeric value associated with it: the value 4 is associated with read permission, the value 2 is associated with write permission, and the value 1 is associated with execute permission. If a permission is denied, then its value is always zero. (In the example above, all permissions have been granted.) For each category of user (owner, group member, or other) these three permission values potentially add up to seven. If we deny one or more type of permission, then that value (4, 2, or 1) is subtracted from the value for that category of user. Thus, if we wish to deny write permission to to the owner's group, we subtract 2 from the total of that permission, which leaves a column value of 5. And if we wish to deny both write and execute permissions to "others," we subtract both 2 and 1, leaving a value of 4. These changes are shown in the array below:

	owner	group	other
read	4	4	4
write	0	0	0
execute	1	1	1
total "value":	7	5	4

The total value is now 754 rather than 777. Note that whatever combination of permissions we create, the numbers will always be a unique representation o that combination, as shown in the following chart:

column	permissions
0	no permissions
1	execute only
2	write only
3	execute and write
4	read only
5	read and execute
6	read and write
7	read, write, and execute

Total Value

Just as each column designates a specific combination of permissions, so the total value represents a specific combination of permissions associated with user types, since the order is always given as: owner--group--other. Thus, from any three digit total value, you can deduce each of the nine possible permissions. This three-digit "total value" (in the examples above, 777 and 754) is used in defining and changing permissions, as described below.

Remember that this total value is always given in the order:

owner--group--others.

Setting Permissions

When you wish to set the mode of a file (set the permissions) you use the **chmod** command at the system prompt (% or $), as in the following example:

```
% chmod 644 myfile
```

This command designates that the file named "myfile" has read and write permission for the owner (you), and read only permission for the group and others. Remember that the permissions for "owner" are always first and the permissions for "other" are always last. Setting permissions for a directory follows exactly the same procedure; you would simply substitute the directory name for the file name.

Paths and Permissions

In order for you to be able to set permissions for a file or directory, Unix must first be able to find the file or directory. Thus, if you are not in the directory that contains the file or directory for which you are setting permissions, you must provide a path name. For example, if you were in your work directory and you wished to set permissions for a file called "file.txt" in a directory called "files" located in your work directory, you would use the following command:

```
% chmod 644 /files/file.txt
```

Determining Current Permissions

To determine the current permissions for a file or directory, use the ls command with the -l (lowercase "L," not a "1") option, as in the following example:

```
% ls -l myfile
```

At the left of the resulting line of output will be the list of permissions expressed as a series of ten letters and hyphens. The last nine spaces are divided into groups of three, each of which will have, in order, an r (read), w (write), and x (execute) or, if that permission has been denied by the file owner, a hypen (-) in its space. As in setting permissions, the three groups of three are given in the order: owner--group--other. For example, the file whose mode was set above as 644 would have the letters:

```
-rw-r--r--
```

This sequence shows that "myfile" is an ordinary file (the first dash; if "myfile" were a directory, there would be a "d" in this location) with read and write permission for the owner (rw-), and read only permission for the owner's group (r--) and for others (r--). If we change the mode again using the command

```
chmod 765 myfile
```

then the "ls -l myfile" command would show as the permissions:

```
-rwxrw-r-x
```

Naturally, only the owner can modify the permissions for a file or directory.

Directory vs. File Permissions

Unix is a "top-down" environment. This means that if you deny "group" or "other" permissions to a directory, all subdirectories and files within that directory will be denied the permissions established at the directory level. You can, of course, grant permissions at the directory level and withhold them at the subdirectory and file level.

Using the Letters

chmod (change mode) Changes file permissions

chmod u+w *file1*	adds write permission for the user to file1
chmod u+rw *filenames*	(sets your own permissions to both read and write)
chmod g+x *file1*	adds execute permission for the group to file1
chmod g+r *filenames*	(allows a group to read these files, but not write to them)
chmod o-r *file1*	removes read permission for the world (other) from file1

chmod o-rwx *filenames* (denies all access to other people on the system)

Note: do not remove the `**x**' from your own permissions for your directories. Think of directories in UNIX as little programs which move you from place to

The following information is from the Unix man pages and describe the chmod function in detail:

CHMOD(1V) USER COMMANDS CHMOD(1V)

NAME

chmod - change the permissions mode of a file

SYNOPSIS

chmod [-fR] mode filename ...

SYNOPSIS

/usr/5bin/chmod [-fR] mode filename ...

AVAILABILITY

The System V version of this command is available with the System V software installation option. Refer to Installing SunOS 4.1 for information on how to install optional software.

DESCRIPTION

Change the permissions (mode) of a file or files. Only the owner of a file (or the super-user) may change its mode.

The mode of each named file is changed according to mode, which may be absolute or symbolic, as follows.

Absolute Modes

An absolute mode is an octal number constructed from the OR of the following modes:

400	Read by owner.
200	Write by owner.
100	Execute (search in directory) by owner.
040	Read by group.
020	Write by group.
010	Execute (search) by group.
004	Read by others.
002	Write by others.
001	Execute (search) by others.
4000	Set user ID on execution.
2000	Set group ID on execution (this bit is ignored if the file is a directory; it may be set or cleared only using symbolic mode).
1000	Sticky bit, (see chmod(2V) for more information).

Symbolic Modes

A symbolic mode has the form:

[who] op permission [op permission] ...

who is a combination of:

u	User's permissions.
g	Group permissions.
o	Others.
a	All, or ugo.

If who is omitted, the default is **a**, but the setting of the file creation mask (see umask in sh(1) or csh(1) for more information) is taken into account. When who is omitted, chmod will not override the restrictions of your user mask.

op is one of:

+	To add the permission.
-	To remove the permission.
=	To assign the permission explicitly (all other bits for that category, owner, group, or others, will be reset).

permission is any combination of:

r	Read.
w	Write.
x	Execute.
X	Give execute permission if the file is a directory or if there is execute permission for one of the other user classes.
s	Set owner or group ID. This is only useful with u or g. Also, the set group ID bit of a directory may only be modified with `+' or `-'.
t	Set the sticky bit to save program text between processes.

The letters **u**, **g**, or **o** indicate that permission is to be taken from the current mode for the user-class.

Omitting permission is only useful with `=', to take away all permissions.

Multiple symbolic modes, separated by commas, may be given. Operations are performed in the order specified.

SYSTEM V DESCRIPTION

If who is omitted in a symbolic mode, it does not take the file creation mask into account, but acts as if who were a.

OPTIONS

-f	Force. chmod will not complain if it fails to change the mode of a file.
-R	Recursively descend through directory arguments, setting the mode for each file as described above. When symbolic links are encountered, their mode is not changed and they are not traversed.

EXAMPLES

The first example denies write permission to others, the second makes a file executable by all if it is executable by anyone:

```
example% chmod o-w file
example% chmod +X file
```

SEE ALSO

csh(1), ls(1V), sh(1), chmod(2V), chown(8)

Chapter 3
Web Page Graphics

Graphics on the World Wide Web

Graphics can add pizzazz and polish to your web pages. In addition to the extra snap your pages will have, however, they will also have an increased download time. This chapter is intended to help you incorporate graphics in your pages and to assist you in using the graphics more effectively.

You have many graphics programs to choose from, but in this handout, we will be referring to Lview, a shareware program available from the UNC-Chapel Hill's ATN shareware archive: http://shareware.unc.edu

Types of Graphic Files

The two graphic file types most commonly found on the World Wide Web are GIF and JPEG (*.gif and *.jpg). The features of both file types are listed below.

1. GIF (Graphics Interchange Format)

- Can be transparent or opaque.
- Supports 256 colors.
- Best for flat color images

2. JPEG (Joint Photographic Experts Group)

- Best for continuous-tone images (saved at medium to low quality for the web).
- Supports millions of colors.

Image Transfer Time

A web page author needs to provide an image that is acceptable in both quality and download time. When including images in World Wide Web pages, consider the intended audience. What type of internet access and hardware are your end users most likely to have? How much time do you think they might be willing to spend waiting for an image to download? Three factors affect the image transfer time: the transfer rate, the file size, and traffic on the network.

Tips For Reducing File Size, and Therefore Download Time

Reduce resolution:	Images for Web display only should be no more than 72 dpi (dots per inch).
Cropping:	Eliminate extra borders and the image area by cropping. Keep only the essential subject matter.
Reduce color depth:	Images meant to be displayed on the Web only rarely need to be 24 bit (millions of colors). Reducing an image from 16 bit color (thousands of colors) to 8 bit (256 colors) eliminates 50% of the image data.

Is color necessary? Often an 8 bit grayscale image (256 shades of gray) will work as well as a color image.

Obtaining Images to Include in Web Pages

Where will you get images for your Web pages? You can use clipart from other programs, create original graphics, scan an image, or use graphics provided at clip art Web sites.

Clipart From Other Programs

Before trying to put a clip art graphic on your Web page, determine the file type of the clipart image. For example, the clipart from Microsoft Word may be in the .wmf format (under Windows 3.x and 95) or PICT format (under Macintosh). Before you can use these images on the Web, you must follow these steps to convert the file to .gif or .jpg:

1. Open the file with your graphics package, using one of two methods: If the file type is not compatible with the graphics software, make it compatible before opening it. Open the file in its native program, then copy and paste it into the graphics software. This may require some experimentation. For example, to use a clipart file from Microsoft Word under Windows 95, follow these steps:
    - Insert the clipart into a Word document first.
    - Click on the graphic in the document, and copy it.
    - Next, paste the graphic into the Paint accessory application and save the file as a bitmap (*.bmp).
2. You can now open the file with LView (select the Open command from the File menu.) If the file type is compatible with the graphics software, you can open it directly within the graphics software by choosing the File menu, and then Open. Select the file by navigating through your drives and folders.
3. With the graphic open in the graphics program (LView, for example), choose **Save As** from the **File** menu.
4. Save the file as either **.gif** or **.jpg** format.

Creating Original Graphics

Graphics for the web may be created in a number of graphic software programs. If saving the file as a .gif or .jpg is an option after creating the graphic, then the file is web-ready. If the file cannot be saved as a .gif or .jpg file, then it must be converted using the steps listed above.

Scanning

Types of Scanners

Many different scanners are available, but you'll use one of two basic types:

- Flatbed scanner
- Slide scanner

The Process

There are different scanning software packages available, so you may need to find specific documentation for the software you are using. Below is a list of general steps you will take to scan an image:

1. Place the original on the scanner.
2. Activate the scanning software. This will differ from scanner to scanner, so be sure to ask for assistance when using a scanner for the first time.
3. If it is possible within the scanning software, scan the image as close to the desired final size as possible.
4. Resolution should be set at 'screen' resolution. The process for doing this will differ from scanner to scanner.
5. Scan the original.
6. If the scanner supports .gif or .jpg files, save the file as one of those two file types (depending on the original).

Scanning at NIEHS

There is a “public use” flatbed scanner (Macintosh platform) with Photoshop and OCR software in A258. Ask your Computer Support Person if your division/branch/lab has their own scanner for internal use.

Graphics from Other Web Sites

When copying an image from the Web, make sure you go to a Web clipart site, and use the images there in accordance with the regulations posted at that site. Otherwise, you will very likely be violating copyright laws.

Including graphics from other web sites is simple.

1. First, use your mouse to point at the graphic you want to copy.

2. Next, if you're using a PC, click with the right mouse button, or, if you're using a Macintosh, hold the mouse button down.

3. This will display a pop-up menu from which you select Save this image as and then release the mouse button. Specify where you want the file to be saved.

HTML Tags for Putting Graphics On Your Page

To include a graphic in your web page, use this tag: **<img src="filename.gif">**

The following table describes attributes you can use in your image source tag:

Attribute	Examples	Description
alt=""	<img src="filename.gif”>	Tells text browsers to display the text shown in quotation marks. If you put nothing in the quotation marks, the text browser displays no reference to the graphic.
align=	<img src="filename.gif" align=right>	Aligns the image to the right (or left) of the page.
border=	<img src="filename.gif" border=0>	Specifies the border size in pixels. The graphic in the example given has no border.

Chapter 4
Web Page Advanced Design

The Basic Tools for Fine-Tuning Your Page: Attributes

Much of what you'll learn in this document involves adding "attributes" to tags you're already using. Attributes allow you to customize certain HTML tags to achieve specific formatting effects: to control alignment and spacing, for instance, or to manipulate text size. Once you've mastered these options, you'll have better control over the "look" of your page. Some attributes require you to use values with them. For spacing, you might specify the exact number of pixels, or a percentage value; for alignment, a directional or positional "value"; for size attributes, your options may be either numeric or relational values. You'll see examples of attributes and values in the rest of this document.

Special Formatting for Graphical Browsers

Most graphical browsers, such as Netscape, can display special formatting like different font and horizontal rule (line) sizes, optional bullet and number types within lists, special table formatting, and text, link, and background colors. If you use this formatting, keep in mind that text-only browsers do not display these special effects, nor are all special formatting attributes recognized by all graphical browsers. Some of the tags and attributes we'll discuss in this document only work with Netscape version 1.1 or higher. We include these here because, in many cases, the functions performed by these Netscape extensions to HTML 2.0 (the current standard) will be included in HTML 3. There is no guarantee, however, that HTML 3 will use the current Netscape tags for these functions--one more reason why Web authors need to keep abreast of new developments.

Adding Comments to Your Page

To make Web pages easier to trouble-shoot and modify, Web authors should develop the habit of adding comments to complex sections of their pages. For example, it's a good idea to label the beginning and end of a table (tables are discussed la ter in this document.) The standard format for comments is:

<!--comment-->

In addition to announcing complex sections, comments can remind you of changes you want to make to your page in the future. For example, you might have a comment like this:

<!--I'm going to put a sound file here as soon as I find it.-->

Putting comments in a consistent format makes it easy to find them all later.

Although comments do not display on your Web page, keep in mind that they are available to the public: anyone who visits your page on the Web can also look at your source code and view your comments.

Controlling the Look of Text

Alignment

The default alignment for text is flush left. To center text, you have several options. Most browsers will recognize the align=center attribute added to the basic paragraph tag set, like this:

** text **

Or, you can use ** ** tag set, which is an HTML extension devised by the folks at Netscape, and is specific to that browser. A third option is to design a borderless table and center your text within the center table cell. For right alignment, use tabs and preformatting, or use table formatting to place text in a right-aligned table cell. (Tables are discussed later in this document).

Changing Font Sizes

If you are designing your page for Netscape, you can control relative font sizes with font-specific tags. Specify the font size used throughout the document with the <basefont> tag (no end-tag required). The basefont default, which is whatever the reader has set as his/her browser preference, is assigned a value of 3.

To increase or decrease the basefont size for your document, add a tag in this format: **<basefont size = value>**, where value is a whole number from 1-7. The size value you choose will adjust the font size relative to the default value of 3: a value of 2 will make your text one font size smaller, while a value of 6 will make it three sizes larger. Since these are relative values, you may have to do some fine-tuning to achieve the effect you want. Remember that most folks use Times 12 point as their default text.

Netscape also allows you to control the font size of particular text sections, paragraphs, lines--even individual characters--with the **** tag set. Use this format:

Text

where value is either: A whole number between 1 and 7 (remember: the default value is 3) + or - (plus or minus) followed by a whole number between 1 and 6, specifying a font size adjustment relative to the base font. For example, **<fontsize = +2>** means the font is two sizes bigger than the base font.

You'll notice that font sizes in HTML are not measured in points. Experiment to find the best font relationships for your page. You may even want to set up a style sheet of text and headings (<h2>, for example) in all font sizes before you use them on your page.

Made to Order: Tailoring Your Ordered and Unordered Lists

Refer to Chatper 1: Web Page Authoring Basics for an introduction to ordered and unordered lists. Netscape extensions allow you to control numbering in ordered lists

(**<ol>**) with the type attribute and different letter values associated with it. For example, to use uppercase letters in your ordered list (ABC), start the list with the tag **<ol type = A>**. Other values for type in ordered lists are a (lowercase letters), **I** (uppercase roman numerals), **i** (lowercase roman numerals), and **1** (arabic numerals, the default type).

You can also use the type attribute to control the bullet shape in unordered lists (**<ul>**). To specify circular bullets in an unordered list, start the list with the tag **<ul type= circle>**. Use the same format to make your bullets **disc** or **square**.

Creating Tables in HTML

If you set up a table in your word-processor using tabs between items, you can pre-format it (**<pre></pre>**) and put it right on your Web page. People with either text-only or graphical browsers can then see your table. Since pre-formatted text appears in a plain fixed font, however, you may decide you want your table to be nicer looking. To create more elegant tables for graphical browsers to read, you can use special table tags and attributes.

Note: The table element is another Netscape 1.1 extension to HTML, but some other browsers recognize it. Tables will be included as a standard in HTML 3.

Define your table with the **<table></table>** tag set. In between these tags, you will also have to specify each particular element of the table and its contents, from rows to captions to individual cells. Use the following tags to define these details:

To specify this element:	**Use these tags:**	**And keep these points in mind:**
Row	**<tr>**row details**</tr>**	Use a set of row tags for each row in the table.
Cell	**<td>**table cell details**</td>**	Use cell tags for each cell in the table.
Heading	**<th>**column or row heading**</th>**	You can have column or row headings. Default is formatted as bold and centered.
Table title	**<caption>**Table Title**</caption>**	You can control the caption style with heading tags: <caption> text </caption>

Here's an example of an extremely simple table:

*<table> <caption><h2>*Some ACC Rivals*</h2></caption> <tr>*
*<th>*University*</th> <th>*City*</th> <th>*Mascot*</th> <th>*Team Name*</th> </tr>*
*<tr><td>*UNC-CH *</td> <td>*Chapel Hill*</td> <td>*Ram*</td> <td>*Tar Heels*</td></tr>*
*<tr> <td>*Duke*</td> <td>*Durham*</td> <td>*Blue Devil*</td> <td>*Blue Devils*</td>*
*</tr> <tr> <td>*NCSU *</td> <td>*Raleigh*</td> <td>*Wolf*</td> <td>*Wolfpack*</td></tr>*
</table>

which looks like this:

Some ACC Rivals			
University	City	Mascot	Team Name
UNC-CH	Chapel Hill	Ram	Tar Heels
Duke	Durham	Blue Devil	Blue Devils
NCSU	Raleigh	Wolf	Wolfpack

If you look at this ACC rivals table with a browser, you'll probably want to make it look better. To improve the look of your table, add attributes to the tags. For example, to put a border around the table, add the border attribute to the table tag, this way: <table border>. The following table lists several attributes and how to use them:

To control this formatting:	**Add this attribute:**	**To one of these tags:**	**And add these values:**	**Examples:**
Border	border	<table>	Border width in pixels (Default with border attribute is 1 pixel.)	<table border> or <table border = 6>
Table width	width	<table>	Number of pixels, or percentage value (not all browsers)	<table width=440> or <width=60%>
Horizontal alignment within an element	align	<tr>, <th>, <td>	left, center, right, justify, decimal (for caption, default horizontal alignment is center)	<td align = right> or <tr align = center>
Vertical align-ment within an element	valign align	<tr>, <th>, <td> <caption>	top, middle, bottom, baseline top, bottom	<tr valign = baseline> or <th valign = top> <caption align = bottom>
The width of a cell, measured in columns	colspan	<th>, <td>	Number of columns (default = 1)	<td colspan = 2>
The height of a cell, measured in rows	rowspan	<th>, <td>	Number of rows (default = 1)	<th rowspan = 3>

As a practice exercise, try adding some of the above attributes to the North Carolina Rivals table example presented earlier. For more table attributes and information, go to the Tables in Netscape 1.1 page at:

http://home.netscape.com/assist/net_sites/tables.html

and the Table Sampler page at

http://home.netscape.com/assist/net_sites/table_sample.html

You may have already realized that creating tables with table tags can be time-consuming. You'll probably decide to use them sparingly.

Controlling Horizontal Rules

The basic tag for a horizontal rule (line) is <hr> (no end-tag). You can add attributes to the following features of horizontal rules:

Thickness.	Use the size attribute together with the rule size in pixels; for example, <hr size = 6> makes a nice fat rule.
Width across the page.	Use the width attribute together with either the rule's width in pixels, or a percentage of the document's width. Include a percent sign (%) for percentages.
Alignment.	Use the align attribute with the value left, right, or center.
Fill.	Change the rule from shaded to solid with the noshade attribute.

Here's an example of a horizontal rule with special formatting:

<hr size=6 width=50% align=right noshade>

which produces this line:

Changing Text Color

To change the color of text, add the text attribute and a color value to the <body> tag at the beginning of your document, like this:

<body text ="#rrggbb">Your entire page</body>

Instead of #rrggbb, put in a color value, which you can select from the Background Colors page:

http://www.infi.net/wwwimages/colorindex.html

For example, if you want your page to have bronze-colored text, add this tag to the beginning:

<body text ="#8C7853">The Illustrated Story of My Life</body>

You can also change the color of the three different types of links by adding several Netscape extension attributes to the <body> tag:

- To change the color of a link before you follow it, use **link**
- To change the color of the link after you visit it and return, use **vlink**
- To change the color of the active link, that is, the color of the link as you click on it, use **alink**

Here's an example that shows text and link colors:

> *<body text="#00FFFF" link="#FFFF00" vlink="#FFFFFF" alink="#FF00FF">*
> My life really changed when I discovered **World Wide Web authoring**. I wanted to read **everything I could find about HTML**.
> *</body>*

If you look at that text with a browser, you'll see that it's not easy to read. The text doesn't contrast enough with the background. Let's look at how to fix that next.

Changing Backgrounds

Specify a background color with the bgcolor attribute in the <body> tag. This attribute is another Netscape 1.1 extension. To change the background to a solid color, use the same type of color codes you use for text and link colors. If you want to make a black background, for example, you'd use this tag: <body bgcolor= "#000000">. Try adding that background to the text color example used earlier.

Instead of a solid color, you might want to use a pattern or other graphic as your background. You can find an assortment of background graphics at this Background Sampler page:

> **http://home.netscape.com/assist/net_sites/bg/backgrounds.html**

Better yet, feel free to design your own. To use graphics as backgrounds, add the background attribute to the <body> tag, and include the location of the graphic file. Here's an example:

> **<body background="http://home.netscape.com/assist/net_sites/bg/marble/purpleblue_marble.gif">**

Keep in mind that, by using a background image, you will be slowing down the speed at which any graphical browser will load your page. For the sake of your readers, especially those using modems, try to keep things simple: a JPEG image, for example, will load faster than a GIF because the JPEG format uses less memory (see the section below on Adding Multi-Media).

Combine background, text, and link attributes in the <body> tag to create the page you want. Again, however, be aware of browser differences when using colors or backgrounds images. It's your page, so you don't have to design for the lowest common denominator (in terms of technology), but some of your special effects may be lost on your viewing public. Of course, there's always the disclaimer route: "This page was designed for viewing with [Netscape Navigator] [Mosaic] [whatever] version X.x or above."

Adding Multi-Media to Your Page

Multi-media refers to graphics, sound, video, and animation for your Web page. As we discuss some of the tools you can use to handle multi-media files, keep these points in mind:

As with everything on the Web, tools to create and manipulate Web multi-media files change quickly, and a new tool is introduced almost daily. The programs we discuss here will get you started, but will be obsolete as soon as you discover another, better program. You can download many shareware Web tools from the Shareware.com page.

http://www.shareware.com/

Be sure to read any README files you download with the shareware programs.

In this document, we'll concentrate on how to put graphics on your page, but we'll also take a brief look at other multi-media file types.

Graphics Files for the Web

As you may know, many types of digital graphics are out there: GIFs, tiffs, and bmps (bitmaps) are among the file types you'll hear discussed. When you add graphics to your page, use file types that most browsers can read or that freeware or shareware helper applications can read. For crisp, sharp in-line graphics, we recommend GIF files. For scanned photographs and other realistic images, we recommend either JPEG or GIF. Note: not all browsers can read JPEG in-line images.

To check your graphics with a browser, you may need helper applications. You can download a number of browser helper applications from ATN Shareware at

http://help.unc.edu/shareware/subpages/helperapps.html

How Do I Get Graphics for the Web?

You can convert files, copy them, or create them. We'll discuss converting and copying files; you can also use numerous software packages to create them.

You may need to convert graphics from one file type to GIF or JPEG. Here are two shareware programs you can use to convert existing files to GIFs, or GIFs to another format:

- If you're using a Macintosh, try GIFConverter
- If you're using a PC, try LVIEWPRO

You can copy graphics from other pages on the Web. As a rule of thumb, only copy and use graphics displayed on pages that are set up specifically for that purpose. YAHOO lists several sources on its Clip Art page at

http://www.yahoo.com/Computers_and_Internet/Multimedia/Pictures/Clip_Art/

In Netscape, a pop-up menu allows you to copy or save an in-line Web graphic. You display the menu one of two ways:

- If you're using a Macintosh, click on the image and hold the mouse button down until the menu pops up.
- If you're using a PC, click on the image with your right mouse button.

The steps for saving or copying the file are fairly straightforward once you get the menu.

Making Buttons & Other Graphics into Links

To transform a simple graphic, such as a button, into a link, provide the anchor reference tag first, then the reference to the graphic, or image, like this:

** <img src="your graphics file.gif">**

Don't forget to add a separate text link for text-only browsers. One way to provide the link is to add the alt attribute to the <img> tag, like this:

<img SRC="familyatplay.gif" alt ="I wish you could see what a good looking family I have.">

The link for Text-browsers will be the text that follows alt=.

Positioning the Graphic on the Page

You can align a graphic on your page by adding the align attribute to the <img> tag, as in this example:

<img src = "mydog&me.gif" align=right>

The following table describes other alignment possibilities. Those that are Netscape-specific are marked with an asterisk *:

To align the image. . .	Add this attribute to your <img> tag:
With the left margin with subsequent text wrapping to right of image	align=left
With the right margin with subsequent text wrapping to the left of image	align=right
With the tallest item in the line	align=top*
With the tallest text item in the line	align=texttop*
Middle to the middle of current line	align=absmiddle*
Middle to the baseline of current line	align=middle*
Bottom to the baseline of current line	align=baseline*
Bottom to the bottom of current line	align=absbottom*

Speed Up Page Loading with Sizing Attributes

Graphics can frustrate visitors to your page because they can slow page loading. You can speed things up by specifying the graphic's size in pixels. When you add the width and height attributes to your <img> tag, Netscape inserts a place holder the size you specify, and continues to load the rest of the page as the graphic is loading. Here's an example:

<img height=36 width= 57 src="myfirstgradeteacher.gif">

If the size you specify is different from the image file, Netscape scales the image to fit the placeholder. Size attributes for images are included in the proposed HTML 3 standard.

Combining Attributes to Get Things Just Right

You can combine multiple attributes in the <img> tag. For example, your tag might look like this:

*<img src="Picture 1.GIF" width = 100 height = 75 align=right top alt="*I wish you could see this picture.”>

Special Effects

Making Graphics Transparent

Normally, a graphic's background is like the mat that framers put around a picture, and the mat blocks out your page's background. If you want your page background to be the graphic's background, you can use shareware programs to make the graphic's background transparent. Currently, the ATN Shareware Distribution page offers Transparency for the Macintosh and GIFTRANS for the PC.

Changing How Graphics Display: Interlacing Images

When you load a page, graphics normally begin to display from the top down, one section at a time. When a section displays, it has the best resolution it will ever have. Interlaced images, on the other hand, appear all at once, with imperfect resolution, and gradually the resolution improves. To interlace a graphics file, use one of the GIF conversion programs discussed earlier; you don't need any special tags or attributes to add the interlaced image to your page.

The Wizard of Oz on the Web

Have you seen the movie The Wizard of Oz ? Remember when everything goes from color to black and white? You can achieve the same effect on the Web by combining the image source attribute (src, as in <img src = "x.gif">) with the low resolution attribute (lowsrc). When you combine these attributes, one graphic replaces another over the time it takes both

graphics to load. The attribute combination was designed to slowly replace low resolution graphics, which load quickly, with high resolution graphics, which load more slowly. You can also use them, for example, to change from a black and white to a color graphic, as we might do on a Wizard of Oz page, or from one graphic to another, which you can use for simple animation. Here's how a tag looks using these attributes:

<img src="babypicture.gif" lowsrc="UNC-CHgrad.gif">.

Try it out, and you'll soon find that you're not in Kansas anymore. Note: HTML 3 proposes a combination of the new <fig></fig> and <overlay> tags as an improvement upon the lowsrc Netscape extension. Refer to the HTML Reference Manual URL at the end of this document.

Other Multi-Media: Movies & Sound Files

You need special equipment and software to digitize photographs, movies, and sound files. Computer stores stock cameras, scanners, and software you need to create these multi-media files. If you produce your own files, use the following file formats so that your page is accessible to as many people as possible:

- For audio: .au files. Sound files are played by helper applications, and most of the free/shareware available can handle .au files.
- For video, provide three file types different users can choose from: MPEG (.mpg) for video with no sound: .avi for video with sound: (for Windows users only) QuickTime (.mov) for video with sound (for users of many platforms in addition to Windows)

Even More Possibilities

Web pages give you endless possibilities for creativity. We can't cover everything here, but we wanted you to be aware of a few more options:

Image Maps: An image map is a graphic that has multiple "hot-spots" that link to other locations. To see an example of image-mapping, check out ATN's home page at http://help.unc.edu/ (use a graphical browser). The bar on the left is an image map. To learn how to create image maps, visit the NCSA Imagemap Tutorial at :

http://hoohoo.ncsa.uiuc.edu/docs/tutorials/imagemapping.html

CGI Scripts: Make your pages more interactive with CGI (Common Gateway Interface) scripts. To learn more, visit the CGI page at

http://hoohoo.ncsa.uiuc.edu/cgi/

Note: CGI does involve programming.

Explore the exciting new world that CGI scripts can add to your web pages. The resources at:

http://www.niehs.nih.gov/websmith/webtools/cgiref.htm

will help you to understand and use CGIs. These are programs that reside on the server and manipulate data sent to them by the browser. CGIs are most commonly used to interpret data sent in via fill-out forms.

Please Note: You must contact the NIEHS Web Consultant in order to put a CGI on the NIEHS Web server. There are many programming and security issues that need to be addressed when using CGIs.

Forms with Gform: Even if you're not a programmer, you can use GForm to create Web forms easily. The GFORM page gives more information at

http://www.gu.edu.au/gwis/cwis/gform/cwis_gform.html

Special Characters: You can have special characters on your page, such as À and ®. You'll find the codes for special characters on this page:

http://www.uni-passau.de/~ramsch/iso8859-1.html

HTML editing software: If you don't like using HTML tags and you don't mind giving up some control over your page, you can prepare Web documents with HTML editors that work like word processors. Shareware HTML editors are available from the on the Web.

There's a world of Web authoring help available to you on the Web itself. The most useful help site for HTML we know of is the HTML Reference Manual at:

http://www.sandia.gov/sci_compute/html_ref.html

Of course, you'll continue to learn more about Web page by studying the source code for pages that impress you, and as you gain experience developing your own pages.

Chapter 5
Web Page Frames

Web Authoring: Netscape Frames

Using Netscape Frames

Netscape Communications, the maker of the Netscape Navigator Web browsers, has proposed HTML tags that allow the inclusion of frames into a single document. The HTML frame tags allow a single browser window to be divided into separate regions that can display different pages of text or images simultaneously. Frames are not currently supported by all Web browsers, but it is likely that they will become a standard in HTML code. To accommodate users who do not have a frames-capable browser, it is a good idea to provide versions of your pages that are accessible without frames.

Content Documents Versus Layout Documents

With the incorporation of the frame tags in HTML, the ".html" document type will represent two different types of documents: layout documents and content documents.

- **Layout documents** contain the HTML code that specifies the frame structures for a particular document. The frames declared or defined in these documents will contain references to other documents, the contents of which will be displayed inside the frame.

- **Content documents** are normal HTML documents that contain information such as text, images, and hyperlinks. Content documents can be viewed alone, as the single document in the browser window, or they can be viewed as specified by a layout document. All documents that meet current HTML standards (such as the pages that you have created thus far) fit the definition of content documents.

Basic Frame Layout

In order to get an idea of how frames work, let's have a look at an example of a basic frame layout. The lines are numbered for reference purposes; these numbers should not be included in the actual HTML file.

```
(1)   <html>
(2)   <head>
(3)   <title>Frame Example</title>
(4)   </head>
(5)   <frameset cols=20%,80%>
(6)         <frame src=first.html name=top>
(7)         <frameset rows=*,*>
(8)               <frame src=second.html name=bottom_left>
(9)               <frame src=third.html name=bottom_right>
(10)        </frameset>
(11)  </frameset>
(12)  </html>
```

Frame Declaration Using the <frameset> Tag

As you can see in the above example, the **<frameset>** tag replaces the **<body>** tag that we see in normal content documents. The opening **<frameset>** tag must include an attribute specifying either a row list or a column list. These lists are comma-separated lists of values specifying either row height or column width. Each value that you include in the list will correspond to a frame size in your document. Each value in a row or column list represents the declaration of a frame's existence. These values can be stated in three ways:

1. We can determine the size of the frames by using percentage values for each item in the row or column list. In the example of line 5 of the above code, we have asked the browser to create two frames, the first taking up 20% of the window column space and the second taking up the remaining 80
2. We can also have the browser determine the size of the frames by using the "*" value for each item in the list (as we did in line 7 of the above code). When using the "*" value, the browser will evenly divide the available space for the number of frames that we've declared (e.g., the attribute "rows=*,*" will give us two frames of equal size while the attribute "rows=*,*,*" will give us three frames of equal size).
3. Finally, we can specify pixel values in our row or column lists, for example: <frameset cols=70,100,500>. This column list will create three frames: the first will be 70 pixels wide, the next 100 pixels wide, and the last 500 pixels wide.

NOTE: Using pixels can lead to problems since different browsers have different screen sizes. It's possible to get error messages if pixel sizes are used incorrectly, so you might want to use them sparingly.

These three methods of frame declaration can be used separately or in any combination. For example, the tag **<frameset cols=30%,100,*,*>** will declare one column that takes up 30% of the screen, will declare another that is 100 pixels wide, and will divide the remainder of the space into two columns of equal width.

NOTE: The list of values must be separated only by a comma with no spaces. If you include a space, the frames will not appear.

Frame Definition Using the <frame> Tag

Once we have used the **<frameset>** tag to declare the number of frames we want and where we want them, we must use the **<frame>** tag to tell the browser what to put in these frames. The **<frame>** tags will appear between the opening **<frameset>** and closing **</frameset>** tags. We will use the **<frame>** tag to specify the location of the content document and to assign a name to the content document to allow us target it for updates when a hyperlink is selected. (We'll look at this below in Targeting Frames for Content Update.)

Specifying the Frame's URL

In order for anything to appear in the frames you declared in the <frameset> tag, a URL needs to be listed, identifying a content document for each frame. This can be accomplished by adding the "src=URL" attribute to the <frame> tag. The URL should refer to the location of the content document you intend to appear in this frame. Look at the example in line 6 of the above code:

```
(6)  <frame src=first.html name=top>
```

When loading your page, the browser will first construct the frames that you've created and then load the URLs into the appropriate frames.

Naming the Frame

Assigning names to frames makes it possible to target them for updates when a hyperlink is selected. This way, you can have a hyperlink appear in one frame and have the content document associated with each link appear in another frame when the hyperlink is selected. (The issue of targeting frames will be covered later in this document.) A frame can be given a name by including the "name=frame_name" attribute in the <frame> tag. Again, line 6 of the above code gives an example of this naming convention.

Directly and Indirectly Nested Frames

To create the frame structure that we see in the example listed above in this document, it is necessary for us to use nested frame declarations. Nested frames allow us to have columns within rows and vice versa. A nested frame is formed by splitting one frame, called the parent frame, into multiple frames. The creation of these frames is accomplished in one of two ways:

Directly Nested Frames

Directly nested frames are used in the example at the beginning of this document. Directly nested frames are created by nesting a second **<frameset>** declaration inside the initial opening and closing **<frameset>** tags. The nested **<frameset>**

declaration appears in place of one of the initial **<frame>** tags. See lines 5-11 for an example of a directly nested frame declaration.

Indirectly Nested Frames

Indirectly nested frames are created by using the URL in one of the initial <frame> tags to refer to a layout document rather than a content document. Consider the following:

```
<frameset rows=*,*>
<frame src=a.html>
<frame src=bc.html>
</frameset>
```

This code represents a layout document that will establish two frames with an equal number of rows. The first **<frame>** tag specifies the content document "a.html." The second **<frame>** tag, however, could specify a second layout document (rather than a content document) called "bc.html" that could be used to declare additional frames.

Why point out this distinction between directly and indirectly nested frames? Good question. Both directly and indirectly nested frame look the same as far as the layout is concerned. The difference arises in the assignment of names to frames or collections of frames. Indirectly nested frames use the <frame> tag to refer to another layout document, thus making it easier to assign a name to a collection of frames.

Targeting Frames for Content Update

Normally, the selection of a hyperlink in an HTML document makes the current document disappear from the browser window in order to display the contents of the next document. With frames, however, we have the option of having the browser update the frame containing the hyperlink, a different frame, a set of frames, or the entire window.

Why would we want to do this? Consider a situation where we provide links to pages outside of our domain. When browsers follow these links, we have no way of controlling whether or not they return to our pages. With frames, however, we can allow our browsers to view the content of other people's pages without ever having them leave our domain. The contents of other people's pages can be loaded into one of our frames.
We can determine where a content document is going to appear by using the frame name to target the destination. The format of a hyperlink tha targets a frame appears below:

text for hyperlink

Target Names

When frames target other frames for update, they share a special parent-child relationship. A child frame is the frame containing the content document dislayed from a hyperlink in another frame. The frame containing the hyperlink is the parent frame.

The target name may be an explicit name given in the <frame> tag (see lines 8 and 9 of the first code). It can also be an implicit name determined by the target frame's relationship to

other frames. These names all start with the underscore (_) character. Some implicit names are listed below:

_self : This name tells the browser to update the frame containing the selected hyperlink. This is the default target that will be selected if no other target is specified.

_parent : If a hyperlink using the "target=_parent" attribute appears in a child frame, the content document for child document's link will appear in the frame with the link that first referred to the child document's frame.

_top : This name instructs the browser to update the entire window, regardless of the frame layout.

_blank : This name tells the browser to display the document's content in a new, untitled window.

Don't be concerned if these implicit target names seem a bit convoluted and confusing. You can avoid them altogether by explicitly naming our frames in the **<frame>** tag and using those names when you target your hyperlinks. Let's take a look at an example.

Layout Document:

*<html> <head> <title>*Layout*</title> </head>*
<frameset cols=15%,85%>
<frame scrolling=no src=x.html>
<frameset rows=90%,10%>
<frame src=a.html name=top_right>
<frame src=b.html>
</frameset> </frameset>

The first frame defined in the above code will serve as the location for our table of contents, named x.html. The rest of this code defines the remaining layout for our frames. We are using a directly nested **<frameset>** declaration to define two frames to appear to the right of our first frame (the one containing the table of contents document). For now we will only be targeting the first of these frames, called a.html, so it is the only one we've given a target name. Below is the code for the two files, x.html and a.html.

x.html	a.html
<html>	<html>
<head>	<head>
<title>X</title>	<title>A</title>
</head>	</head>
<body>	<body>
X	Text for links to the left will page
</body>	</body>
</html>	</html>

In this example, **x.html** serves as our table of contents and **a.html** is the default content file in the viewing location for our links. You see that x.html contains one link, a reference to a fictitious contents page called page.html. If we create this page, when we select the link for it, the contents of page.html would appear in the frame that currently holds the content for a.html. Try creating these files to see how this works.

Final Frame Suggestions

Below are a number of potential pitfalls when using frames as well as a summary of frame tags.

Improper Row and Column Lists

Improper row and column lists are lists that do not specify valid sums of row heights or column widths, for example:

<frameset rows=30%,40%,50%>

This list is improper because it specifies a sum of column widths that exceeds 100%. Another potential improper list involves the use of pixel values. Consider this example:

<frameset cols=25,40,100>

This declaration tells the browser to create one column 25 pixels wide, another 40 pixels wide, and a third 100 pixels wide. This declaration is not (usually) acceptable because it assumes the total available space is 165 pixels wide. Problems will ensue if the total available area is wider or narrower than 165 pixels.

Frame-Incapable Browsers and the <noframes> Tag

What do you think will happen if someone tries to view your frame layout document in a browser that does not support frames? He or she will see a whole lot of nothing. Since layout documents contain no text, the browser will load a blank browser screen. We can avoid this problem by implementing the <noframes> tag. Frame-capable browsers will ignore any text that appears between the opening and closing <noframes> tags, so they can be used to provide information to people using frames-incapable browsers.

Summary of Frame Tags and Attributes

Tag	Attributes	Values	Description
<frameset> </frameset>	rows	Pixels, %, *	Declares the number and size of the horizontal frames into which you want the screen to be divided.
	cols	Pixels, %, *	Declares the number and size of the vertical frames into which you want the screen to be divided.
<frame>	src	URL	Specifies which document to display in the frame.
	name	Name of frame	Assigns a name to the frame so it can be referenced by other frames.
	noresize		Prevents the browser from resizing the frame.
	scrolling	Yes, No, Auto	Specifies if scroll bars should definitely appear, never appear, or appear only if needed.
	marginheight	pixels	Specifies the top and bottom margins of the frame.
	marginwidt h	pixels	Specifies the left and right margins of the frame.
<noframes> </noframes>			Tells frames-capable browsers to ignore text and tags between the <noframes> tags, so programmers can put information for non-frames capable browsers.
<a>	href target	URL name	Makes a hyperlink. Allows a link to specify which frame to display the linked document in. The name is the name specified in the <frame> tag.
	_self		Indicates the new document should be displayed in the same windows as the link.
	_parent		Puts the new document in the parent frame of the frame containing the link.
	_top		Removes frames from the screen and displays the document in the whole window.
	_blank		Puts the new document into a new, untitled window.

Chapter 6
Web Page Forms and Counters

Creating Forms Using gform

From time to time, we may want to ask a few questions of the people who are looking at our web pages and examine the answers that they return. Previously, we could only accomplish this goal by including an anchor, <a href="mailto:email@server">, to receive email input. Now however, we can create forms in our Web pages which allow people to submit information. The form method that we will be looking at here is called gform. The gform method was created by Larry Raaymakes and Jonathan Magid.

With gforms, we can design our own forms and specify how the data should be returned. We can have the gform submit the data in the form of an email message or we can have it append the data to a preexisting data file. Let's look at an example of a form using gform to examine its construction piece by piece. (Each line shows the line number in parentheses. This number is not meant to be included in the source of your HTML file.)

The <form> Tag

The <form> tag declares the existence of a form.

(1) <html>
(2) <head>
(3) <title>Forms with gform</title>
(4) </head>
(5) <body bgcolor="#FFFFFF">
(6) <form method="post" action="/cgi-bin/gform?
/home/userid/directory/filename.html">

The initial tags used in this file are exactly the same tags that you would use to create any other Web page. The only unfamiliar bit of information that we see is the **<form>** tag. There are two attributes here of which we must be aware: the method attribute and the action attribute. The "**method=**" attribute tells the form to post the data that it receives in one of two ways. Use the "post" value here. We'll see why a little later. The "**action=**" attribute specifies what program to use to process the data received in the form (**/cgi-bin/gform?**) and the location of the file that displays the form (**/home/userid/directory/filename.html**).

Input Types

Forms created using gform can support a number of different types of input. We receive input in our forms by using the <input> tag. We'll look at each of these different types of input in the following sections.

Text Input Type

(7) <*h4*>Text Input<*/h4*>
(8) First Name:<*INPUT TYPE=text NAME="first" SIZE=15*><*br*>
(9) Last Name:<*INPUT TYPE=text NAME="last"SIZE=15*><*br*>
(10) <*p*>

The first, and most important, attribute that we see added to the **<input>** tag is the type attribute. This attribute tells the browser what type of input we are expecting and instructs it how to construct the input box. The name attribute is a variable that will be used later when we decide how we want to receive the data. We'll look more at this later. The size attribute tells the browser how many characters long you want the input box to be.

Checkbox Input Type

(11) <*h4*>Checkbox<*/h4*>
(12) These are my favorite schools:<*br*>
(13) <*dl*>
(14) <*dd*><*INPUT TYPE=checkbox NAME="favs" VALUE="UNC-CH" CHECKED*>UNC- CH <*br*>
(15) <*dd*><*INPUT TYPE=checkbox NAME="favs" VALUE="UCLA"*>
UCLA<*br*>
(16) <*dd*><*INPUT TYPE=checkbox NAME="favs" VALUE="Duke"* >Duke<*br*>
(17) <*dd*><*INPUT TYPE=checkbox NAME="favs" VALUE="NCSU"*>
NCSU <*br*>
(18) <*/dl*>
(19) <*p*>

With the checkbox input type, we have a list of items that will appear in our form with a small checkbox in front of the text. We can "check" as many of the boxes as we like. Notice the "name" variable that we assign to each of these input boxes is the same. We are using the same value here because our respondents can choose one or many of these choices. By giving them all the same "name" variable, we can see all that they checked when we receive the data. The "**value**" attribute tells the form what to list when we receive the data. Here we want to use names that uniquely identify each item. The "checked" attribute tells the browser which one to initially select (i.e., when the page loads, this item will already have a check in its box).

Radio Buttons

(20) <*h4*>Radio Buttons<*/h4*>
(21) <*dl*>
(22) <*dd*><*INPUT TYPE=radio NAME="best" VALUE="UNC-CH CHECKED*>
UNC-CH <*br*>
(23) <*dd*><*INPUT TYPE=radio NAME="best" VALUE="UCLA"*>UCLA<*br*>
(24) <*dd*><*INPUT TYPE=radio NAME="best" VALUE="Duke"*>Duk*e*<*br*>
(25) <*dd*><*INPUT TYPE=radio NAME="best" VALUE="NCSU"*>NCSU <*br*>
(26) <*/dl*>
(27) <*p*>

Radio buttons are similar to the checkbox type of input. With radio buttons, however, only one item can be selected at a time. Otherwise, the attributes are the same as the checkbox example.

Drop-Down Box

(28) <*h4*>Select</*h4*>
(29) On Saturdays, I usually play: <*SELECT NAME="sat"*>
(30) <*OPTION SELECTED*> soccer
(31) <*OPTION*> basketball
(32) <*OPTION*> football
(33) <*OPTION*> ultimate frisbee
(34) <*OPTION*> the cello
(35) </*SELECT*><*br*>
(36) <*p*>

The **<select>** tag creates a drop-down box, which is similar to the "radio" type of input in that we are offered a number of items from which we can choose only one. The display of these items, however, is significantly different. By using the **<option>** tag, we can enter our list items which will appear as a single input box with a drop-down list of items from which the user can choose only one. The **<option>** tag with the selected attribute will appear first in the text box when the page is loaded in the browser window. Again, the name attribute will be used later when we receive our data.

Text Area Input

(37) <*h4*>Text Area</*h4*>
(38) This is what I did yesterday:
(39) <*TEXTAREA NAME="yesterday" ROWS=12 COLS=50*></*TEXTAREA*>
(40) <*p*>

The **<textarea>** tag is used to create a text box in which users can type free text. The rows attribute determines how many rows high the text area will be. The **cols** attribute determines how many characters wide the text area will be.

Now What? Form Submission

Submit or Restart

(41) <*INPUT TYPE=submit VALUE="Submit"*>
<*INPUT TYPE=reset VALUE="Start over"*>
(42) </*form*>

These two input types will create buttons that give people the opportunity to submit their answers or to clear the form and start over again. The **type** attribute tells the browser which button will be used to submit the form and which will be used to reset it. The text value specified in each of the value attributes will appear as the name of each button. We can use any value that seems appropriate to us. Finally, we are ready to close the form using the **</form>** tag and we can move on to the display of our data.

Displaying and Receiving Results

```
(43) <p>
(44) <!gform "$(first)">
(45) <!gform "$(last)"\n"
(46) <!gform "My favorite schools include $(favs),”>
(47) <!gform "although $(best) is the one I like best.\n”>
(48) <!gform "On Saturdays, I play $(sat)\n”>
(49) <!gform "$(yesterday)\n">
(50) <!gform "----------\n">
(51) <!gform DELIVER=mail"userid@server" Subject="form input”>
(52) <!gform REPLY="/home/jennw/forms/thanks.html”>
(53) </body>
(54) </html>
```

This section shows how we will receive our data. Variable names (necessary for the display of our data) are listed in **parentheses** and preceded by the **$** sign. We can print strings of text that aren't variables by putting the text in **quotations** (see example in line 46). The "**\n**" will put a line break in the display of your results and continue printing the data on the following line. The "**deliver**" attribute tells the form program how you want to receive the results.

As Email

We can receive this data in an email message (as shown in line 51) where "**userid@server**" is replaced by a valid email address and the "**subject**" attribute shows the text for the subject line of the mail message.

To a File

We can also have the data appended to an existing text file. Let's look at an example of this:

<!gform DELIVER=file"directory/filename">

The file listed in the attribute above must be a pre-existing file (the form will not create it for you) and must have world-writable permissions set for it.

Responding to Input

The **reply** attribute (line 52) can specify the location of another HTML file that acknowledges the receipt of the form information and thanks the user for their participation. It is not necessary to include this line, but it is a nice touch that lets people know that their form answers were submitted successfully.

Adding a Counter to Your Page

A **counter** tells people how many times the page has been hit (visited). If you are using Jeeves as your Web server, go to the page:

http://www.niehs.nih.gov/websmith/counter.htm

to set up the data file for your counter. This file stores the number of hits. After you fill out the form, you will be given an HTML tag to put in your document where you want the counter. For example:

<IMG SRC="/cgi-bin/Count.cgi?ft=0|frgb=69;139;50|tr=0|trgb=0;0;0|wxh=15;20|md=6|d=C|st=5|sh=1|df=hitcounttxt.dat" align=absmiddle>

This tag looks pretty confusing, but each of the arguments are explained below:

Argument	Name	Description
ft	Frame Width	Specifies the width of the frame, from 0 (none) to 9 (very thick).
frgb	Frame Color	Indicates the color of the frame in RGB format; the red, green, and blue codes are separated by a semi-colon.
tr	Transparency	Turns transparency on (1) or off (0).
trgb	Transparency Color	Indicates what color, in RGB format, should be made transparent.
wxh	Width & Height of the digits	Allows you to specify, in pixels, how high and wide you want the digits to be.
md	Maximum Digits	Specifies how many digits should be shown (5-10).
dd	Digit Type	Specifies the style of the digits. Styles are A-D.
df	Data File	Refers to the data file you set up on the web page given above.

For more information on counter attributes see

http://www.fccc.edu/users/muquit/Count.html.

Chapter 7:
Access Restrictions

Restricting Access to Your Web Pages

Occasionally you will create pages that you don't want to have seen outside of NIEHS. In fact, you may want to have pages that are only accessible to your particular program or group, or that are restricted for only your own eyes. Since we do not have a firewall in place, any pages residing on the jeeves server can be seen by anyone browsing the Web. If you are interested in keeping some of your pages confidential, you will need to create some access restrictions within your directories.

The first step in creating restricted directories is to create a directory that all your restricted files can be saved in. For example, NIEHS policy pages on the Web Consultant pages are stored in a directory called **restrict**. That way, only individuals coming from **niehs.nih.gov** can reference these pages.

The next step is to determine how you want to protect your pages. You can restrict individuals by their **domain**, by **individual password** and by **group passwords**. All three ways are discussed below.

By following the examples set up below, you should be able to create your own set of restricted pages. For more information about each of the commands listed below, check out the **Access Restrictions Glossary** at the end of this chapter. In addition to explaining how each command works, it also includes the general syntax. If you have any problems trying to create restricted directories, please contact the Websmith, for assistance.

Other information sources:

NCSA's description of access restrictions:

http://hoohoo.ncsa.uiuc.edu/

User Authentication by Frostbyte Computer Consultants:

http://www.frostbyte.com.au/auth.html

Restricted domain:

In order to create a domain restricted directory (one in which only individuals at NIEHS can see the pages), you need to create a hidden file called .htaccess. This file needs to be in the directory that you are restricting. Inside this file, you will need the following statements.

AuthUserFile /dev/null
AuthGroupFile /dev/null
AuthName AllowNIEHS
AuthType Basic

<Limit GET>
order deny,allow
deny from all
allow from .niehs.nih.gov
</Limit>

Once this file is added, all subsequent html files in that directory will be restricted. Only the individuals you have given access to will be able to access the files.

Password protected:

In order to create a password restricted directory (one in which only individuals who know the password can see the pages), you need to create a hidden file called **.htaccess**. This file needs to be in the directory that you are restricting. Inside this file, you will need the following statements.

AuthUserFile /usr/local/WWW/html/path to restricted directory/.htpasswd
AuthGroupFile /dev/null
AuthName ByPassword
AuthType Basic

<Limit GET>
require user (your name)
</Limit>

After creating this file, you will need to run a program called htpasswd. This program will allow you to create a password for the username you specified in the .htaccess file. (In the above example (your name) is the username). You will need to telnet to jeeves, switch into the directory that you are restricting, and type,

/usr/local/WWW/bin/htpasswd -c ./.htpasswd username

This will run the program htpasswd and will prompt you for the password you want to use. You will need to enter this password twice. Once you have done this, htpasswd will create a file called **.htpasswd**. This file will be the file that is accessed from **.htaccess**.

Password Protected, Group Access:

In order to create a password restricted directory for a group (one in which only individuals who know the password can see the pages, but one that allows each individual to have a different username and password), you need to create a hidden file called .htaccess. This file needs to be in the directory that you are restricting. Inside this file, you will need the following statements.

AuthUserFile /usr/local/WWW/html/path to restricted directory/.htpasswd
AuthGroupFile /usr/local/WWW/html/path to restricted directory/.htgroup
AuthName GroupByPassword
AuthType Basic

<Limit GET>
require group my-group
</Limit>

After creating this file, you will need to run a program called htpasswd. This program will allow you to create passwords for the group you specified in the **.htaccess** file. (In the above example my-group is the groupname). You will need to telnet to jeeves, switch into the directory that you are restricting, and type:

/usr/local/WWW/bin/htpasswd -c ./.htpasswd username1

After you enter this line, the program htpasswd, will prompt you for a password for this individual. You will need to enter this password twice. Once you have done this, htpasswd will create a file called .htpasswd.

After you have entered the first one, you will use a slightly modified command for the subsequent individuals. Type in the following command (you will do this set as many times as you need to add all the individuals in your group) :

/usr/local/WWW/bin/htpasswd ./.htpasswd username2

The program will prompt you twice for the second individuals password. The password will be added to the already created file, .htpasswd. You will continue until you have entered all the usernames that will be including in your group.

Once you are done adding the usernames and passwords, you will need to create a file called **.htgroup** to the directory you are restricting. This file will be accessed by **.htaccess**. Inside the file **.htgroup** you will need the following statement:

my-users: username1 username2 username3 username4

You will only include as many usernames as individuals you will be including in your group.

Access Restrictions Definitions

AuthGroupFile
Specifies the location and name of the file from which to read groups. You can combine users into groups and allow access by group.

AuthGroupFile filename

AuthName
Specifies the name of the authorization area. This basically assigns a nickname to the current location. This will allow the browser to cache the username and password and resend it automatically - therefore the user doesn't have to re-enter the password information every time he or she backs up one page. This information will also be printed in the dialog box that prompts for the userid and password.

AuthName Area 51
AuthName NIEHS Web Consultant Pages

AuthType
Specifies the method of authentication. Basic is the most commonly used method because it requires no special software on the client. Other authentication types are PEM, PGP, KerberosV4, KerberosV5, and Digest. For purposes of our server, only Basic is used.

AuthType authorization type

AuthUserFile
Specifies the location and name of the file from which to read the password. This will be the UNIX pathname of the password file. Since the password files should be created in the directory that is being restricted, the pathname will always be the current directory.

If there is no password, then this item is specified as /dev/null which stands for "filedoes not exist".

AuthUserFile filename

Limit
Describes the allowed or denied HTTP methods (POST, GET) and remote hosts. This is a directive format that encloses other directives into a block of configuration code. It needs to have beginning and ending tags.

<Limit method>
directives
</Limit>

The directives that are allowed under Limit are:

- *allow from ...*
  Defines a host or hosts from which access is allowed. The host can be an IPaddress, a complete host name, a partial host or all.

allow from host1 host2 ...
allow from all

◊ *deny from ...*
Defines a host or hosts from which access is not allowed. The host can be an IP address, a complete host name, a partial host or all.

deny from host1 host2 ...
deny from all

◊ *order ...*
Defines the order in which the allow and deny statements are processed.

order deny,allow
order allow,deny
order mutual-failure

◊ *require ...*
Defines the users and groups that are required to access this directory.

require user username
require group groupname
require valid-user username

Options
Describes which server features are supported. The default is all.

Appendix A

THE BARE BONES GUIDE TO HTML

by Kevin Werbach

Version 3.0 Formatted -- July 21, 1996

The latest version of this document is available at:
<**http://werbach.com/barebones/**>.

The Guide lists all the tags that current versions of most browsers are likely to recognize. I have included all the tags in the HTML 3.2 specificaıion, as well as Netscape extensions included in versions of Netscape Navigator up to 3.0b5. This Guide is designed to be as concise as possible; more detailed information on HTML is available through my WWW Help Page. Comments and suggestions are always welcome.

Table of Contents

Important: If you are not clear about the differences between HTML 2.0, HTML 3.0, HTML 3.2 and Netscape extensions, I suggest that you read the W3C statement on the development of HTML.

BASIC ELEMENTS

Document Type	`<HTML></HTML>`	(beginning and end of file)
Title	`<TITLE></TITLE>`	(must be in header)
Header	`<HEAD></HEAD>`	(descriptive info, such as title)
Body	`<BODY></BODY>`	(bulk of the page)

STRUCTURAL DEFINITION

Heading	<H?></H?>	(the spec. defines 6 levels)
Align Heading	<H? ALIGN=LEFTICENTERIRIGHT></H?>	
Division	<DIV></DIV>	
Align Division	<DIV ALIGN=LEFTIRIGHTICENTER></DIV>	
Block Quote	<BLOCKQUOTE></BLOCKQUOTE>	(usually displayed as indented)
Emphasis	<EM></EM>	(usually displayed as italic)
Strong Emphasis	<STRONG></STRONG>	(usually displayed as bold)
Citation	<CITE></CITE>	(usually italics)
Code	<CODE></CODE>	(for source code listings)
Sample Output	<SAMP></SAMP>	
Keyboard Input	<KBD></KBD>	
Variable	<VAR></VAR>	
Definition	<DFN></DFN>	(not widely implemented)
Author's Address	<ADDRESS></ADDRESS>	
Large Font Size	<BIG></BIG>	
Small Font Size	<SMALL></SMALL>	

PRESENTATION FORMATTING

	Bold	<B></B>	
	Italic	<I></I>	
N3.0b	**Underline**	<U></U>	(not widely implemented yet)
	Strikeout	<STRIKE></STRIKE>	(not widely implemented yet)
N3.0b	**Strikeout**	<S></S>	(not widely implemented yet)
	Subscript		
	Superscript		
	Typewriter	<TT></TT>	(displays in a monospaced font)
	Preformatted	<PRE></PRE>	(display text spacing as-is)

	Width	`<PRE WIDTH=?></PRE>`	(in characters)
	Center	`<CENTER></CENTER>`	(for both text and images)
N1.0	**Blinking**	`<BLINK></BLINK>`	(the most derided tag ever)
	Font Size	`<FONT SIZE=?></FONT>`	(ranges from 1-7)
	Change Font Size	`<FONT SIZE="+|-?"></FONT>`	(from 1-7; default is 3)
N1.0	**Base Font Size**	`<BASEFONT SIZE=?>`	
	Font Color	`<FONT COLOR="#$$$$$$"></FONT>`	
N3.0b	**Select Font**	`<FONT FACE="***"></FONT>`	
N3.0b	**Multi-Column Text**	`<MULTICOL COLS=?></MULTICOL>`	
N3.0b	**Column Gutter**	`<MULTICOL GUTTER=?></MULTICOL>`	(default is 10 pixels)
N3.0b	**Column Width**	`<MULTICOL WIDTH=?></MULTICOL>`	
N3.0b	**Spacer**	`<SPACER>`	
N3.0b	**Spacer Type**	`<SPACER TYPE=horizontal|vertical|block>`	
N3.0b	**Spacer Size**	`<SPACER SIZE=?>`	
N3.0b	**Spacer Dimensions**	`<SPACER WIDTH=? HEIGHT=?>`	
N3.0b	**Spacer Alignment**	`<SPACER ALIGN=left|right|center>`	

LINKS AND GRAPHICS

	Link Something	`<A HREF="URL"></A>`	
	Link to Target	`<A HREF="URL#***"></A>`	(if in another document)
		`<A HREF="#***"></A>`	(if in current document)
N2.0	**Target Window**	`<A HREF="URL" TARGET="***"||_blank|_self|_Parent|_top|></A>`	
	Define Target in Document Relationship	`<A NAME="***"></A>`	
	Reverse Relationship	`<A REV="***"></A>`	(not widely implemented)
	Display Image	`<IMG SRC="URL">`	(not widely implemented)
	Alignment	`<IMG SRC= H URL" ALIGN=TOP|BOTTOM|MIDDLE|LEFT|RIGHT>`	
N1.0	**Alignment**	`<IMG SRC="URL" ALIGN=TEXTTOP|ABSMIDDLE|BASELINE|ABSBOTTOM>`	
	Alternate	`<IMG SRC="URL" ALT="***">`	(if image not displayed)
	Imagemap	`<IMG SRC="URL" ISMAP>`	(requires a script)

	Client-Side Imagemap	`<IMG SRC="URL" USEMAP="URL">`	
	Map Description	`<MAP NAME="***"></MAP>`	
	Map Sections	`<AREA SHAPE="RECT" COORDS=",,," HREF= "URL"|NOHREF>`	
	Dimensions	`<IMG SRC="URL" WIDTH=? HEIGHT=?>`	(in pixels)
	Border	`<IMG SRC="URL" BORDER=?>`	(in pixels)
	Runaround Space	`<IMG SRC="URL" HSPACE=? VSPACE=?>`	(in pixels)
N1.0	**Low-Res Proxy**	`<IMG SRC="URL" LOWSRC="URL">`	
Nl.l	**Client Pull**	`<META HTTP-EQUIV="Refresh" CONTENT="?; URL=URL">`	
N2.0	**Embed Object**	`<EMBED SRC="URL">`	(insert object into page)
N2.0	**Object Size**	`<EMBED SRC="URL" WIDTH=? HEIGHT=?>`	
	Paragraph	`<P></P>`	(closing tag often unnecessary)
	Align Text	`<P ALIGN=LEFT|CENTER|RIGHT></P>`	
	Line Break	` `	(a single carriage return)
	Clear Textwrap	`<BR CLEAR=LEFT|RIGHT|ALL>`	
	Horizontal Rule	`<HR>`	
	Alignment	`<HR ALIGN=LEFT|RIGHT|CENTER>`	
	Thickness	`<HR SIZE=?>`	(in pixels)
	Width	`<HR WIDTH=?>`	(in pixels)
N1.0	**Width Percent**	`<HR WIDTH="%">`	(as a percentage of width)
	Solid Line	`<HR NOSHADE>`	(without the 3D cutout look)
N1.0	**No Break**	`<NOBR></NOBR>`	(prevents line breaks)
N1.0	**Word Break**	`<WBR>`	

LISTS

Unordered List	`<UL><LI></UL>`	(`<LI>` before each list item)
Compact	`<UL COMPACT></UL>`	
Bullet Type	`<UL TYPE=DISC|CIRCLE|SQUARE>`	(for the whole list)
	`<LI TYPE=DISC|CIRCLE|SQUARE>`	(this & subsequent)

Ordered List	<OL><LI></OL>	(<LI> before each list item)
Compact	<OL COMPACT></OL>	
Numbering Type	<OL TYPE=A|a||I|i|1>	(for the whole list)
	<LI TYPE=A|a|I|i|1>	(this & subsequent)
Starting Number	<OL START=?>	(for the whole list)
	<LI VALUE=?>	(this & subsequent)
Definition List	<DL><DT><DD></DL>	(<DT>=term, <DD>definition)
Compact	<DL COMPACT></DL>	
Menu List	<MENU><LI></MENU>	(<LI> before each list item)
Compact	<MENU COMPACT></MENU>	
Directory List	<DIR><LI></DIR>	(<LI> before each list item)
Compact	<DIR COMPACT></DIR>	

BACKGROUNDS AND COLORS

Tiled Bkground	<BODY BACKGROUND="URL">	
Bkground Color	<BODY BGCOLOR="#$$$$$$">	(order is red/green/blue)
Text Color	<BODY TEXT="#$$$$$$">	
Link Color	<BODY LINK="#$$$$$$">	
Visited Link	<BODY VLINK="#$$$$$$">	
Active Link	<BODY ALINK="#$$$$$$">	

(More info at http://werbach.com/web/wwwhelp.html#color)

SPECIAL CHARACTERS

(these must all be it lower case)

Special Character	&#?;	(where ? is the ISO 8859-1 code)
<	<	
>	>	
&	& amp;	

“	"
Registered TM	®
Copyright	& copy;
Non-Breaking Space	

(Complete list at http://www.uni-passau.de./%7Eramsch/iso8859-1.html)

FORMS

	Define Form	<FORM ACTION="URL"METHOD=GET|POST></FORM>	
N2.0	**File Upload**	<FORM ENCTYPE="multipart/form-data"></FORM>	
	Input Field	<INPUT TYPE=“TEXT|PASSWORD|CHECKBOX|RADIO|IMAGE|HIDDEN|SUBMIT|RESET”>	
	Field Name	<INPUT NAME="***">	
	Field Value	<INPUT VALUE="***">	
	Checked?	<INPUT CHECKED>	(checkboxes and radio boxes)
	Field Size	<INPUT SIZE=?>	(in characters)
	Max Length	<INPUT MAXLENGTH=?>	(in characters)
	Selection List	<SELECT></SELECT>	
	Name of List	<SELECT NAME="***''></SELECT>	
	# of Options	< SELECT SIZE=?></SELECT>	
	Multiple Choice	<SELECT MULTIPLE>	(can select more one)
	Option	<OPTION>	(items that can be selected)
	Default Option	<OPTION SELECTED>	
	Input Box Size	<TEXTAREA ROWS=? COLS=?></TEXTAREA>	
	Name of Box	<TEXTAREA NAME="***"></TEXTAREA>	
N2.0	**Wrap Text**	<TEXTAREA WRAP=OFFIVIRTUALIPHYSICAL></TEXTAREA>	

TABLES

	Define Table	<TABLE></TABLE>
	Table Border	<TABLE BORDER=?></TABLE>

	Cell Spacing	`<TABLE CELLSPACING=?>`	
	Cell Padding	`<TABLE CELLPADDING=?>`	
	Desired Width	`<TABLE WIDTH=?>`	(in pixels)
	Width Percent	`<TABLE WIDTH="%">`	(percentage of page)
	Table Row	`<TR></TR>`	
	Alignment	`<TR ALIGN=LEFT|RIGHT| CENTER|MIDDLE|BOTTOM VALIGN=TOP|BOTTOM|MIDDLE>`	
	Table Cell	`<TD></TD>`	(must appear within table rows)
	Alignment	`<TD ALIGN=LEFT|RIGHT CENTER|MIDDLE|BOTTOM VALIGN=TOP|BOTTOM|MIDDLE>`	
	No linebreaks	`<TD NOWRAP>`	
	Columns to Span	`<TD COLSPAN=?>`	
	Rows to Span	`<TD ROWSPAN=?>`	
N1.1	**Desired Width**	`<TD WIDTH=?>`	(in pixels)
N1.1	**Width Percent**	`<TD WIDTH="%">`	(percentage of table)
N3.0b	**Cell Color**	`<TD BGCOLOR="#$$$$$$">`	
	Table Header	`<TH></TH>`	(same as data, except bold centered)
	Alignment	`<TH ALIGN=LEFT|RIGHT CENTER|MIDDLE|BOTTOM VALIGN=TOP|BOTTOM|MIDDLE>`	
	No Linebreaks	`<TH NOWRAP>`	
	Columns to Span	`<TH COLSPAN=?>`	
	Rows to Span	`<TH ROWSPAN=?>`	
N1.1	**Desired Width**	`<TH WIDTH=?>`	(in pixels)
N1.1	**Width Percent**	`<TH WIDTH="%">`	(percentage of table)
N3.0b	**Cell Color**	`<TH BGCOLOR="#$$$$$$">`	
	Table Caption	`<CAPTION></CAPTION>`	
	Alignment	`<CAPTION ALIGN=TOP|BOTTOM>`	(above/below table)

FRAMES

N2.0	**Frame Document**	`<FRAMESET></FRAMESET>`	(instead of `<BODY>`)
N2.0	**Row Heights**	`<FRAMESET ROWS=,,,></FRAMESET>`	(pixels or %)

N2.0	**Row Heights**	`<FRAMESET ROWS=*></FRAMESET>`	(* = relative size)
N2.0	**Column Widths**	`<FRAMESET COLS=,,,></FRAMESET>`	(pixels or %)
N2.0	**Column Widths**	`<FRAMESET COLS=*></FRAMESET>`	(* = relative size)
N3.0b	**Border Width**	`<FRAMESET BORDER=?>`	
N3.0b	**Borders**	`<FRAMESET FRAMEBORDER="yes|no">`	
N3.0b	**Border Color**	`<FRAMESET BORDERCOLOR="#$$$$$$">`	
N2.0	**Define Frame**	`<FRAME>`	(contents of an individual frame)
N2.0	**Display Document**	`<FRAME SRC="URL">`	
N2.0	**Frame Name**	`<FRAME NAME="***"|_blank|_self _parent|_top>`	
N2.0	**Margin Width**	`<FRAME MARGINWIDTH=?>`	(left and right margins)
N2.0	**Margin Height**	`<FRAME MARGINHEIGHT=?>`	(top and bottom margins)
N2.0	**Scrollbar?**	`<FRAME SCROLLING="YES|NO|AUTO">`	
N2.0	**Not Resizable**	`<FRAME NORESIZE>`	
N3.0b	**Borders**	`<FRAME FRAMEBORDER="yes|no">`	
N3.0b	**Border Color**	`<FRAME BORDERCOLOR="#$$$$$$">`	
N2.0	**Unframed Content**	`<NOFRAMES></NOFRAMES>`	(for non-frames browsers)

JAVA

Applet	`<APPLET></APPLET>`	
Applet File Name	`<APPLET CODE="***">`	
Parameters	`<APPLET PARAM NAME="***">`	
Applet Location	`<APPLET CODEBASE="URL">`	
Applet Identifier	`<APPLET NAME="***">`	(for references elsewhere in the page)
Alternative Text	`<APPLET ALT="***">`	(for non-Java browsers)
Alignment	`<APPLET ALIGN= "LEFT|RIGHT|CENTER">`	
Size	`<APPLET WIDTH=? HEIGHT=?>`	(in pixels)
Spacing	`<APPLET HSPACE=? VSPACE=?>`	(in pixels)

MISCELLANEOUS

	Comment	`<!-- ***-->`	(not displayed by the browser)
	HTML 3.2 Prologue	`<!DOCTYPE HTML PUBLIC "-//W3C //DTD HTML 3.2//EN">`	
	Searchable	`<ISINDEX>`	(indicates a searchable index)
	Prompt	`<ISINDEX PROMPT="***">`	(text to prompt input)
	Send Search	`<A HREF="URL?***"></a>`	(use a real question mark)
	URL of This File	`<BASE HREF="URL">`	(must be in header)
N2.0	**Base Window Name**	`<BASE TARGET="***">`	(must be in header)
	Relationship	`<LINK REV="***" REL="***" HREF="URL>`	(must be in header)
	Meta Information	`<META>`	(must be in header)
	Style Sheets	`<STYLE></STYLE>`	(not yet widely supported)
	Scripts	`<SCRIPT></SCRIPT>`	(not yet widely supported)

Appendix B

WWW Help Page

Written by: Kevin Werbach
Last updated: May 4, 1997

(http://werbach.com/web/wwwhelp.html#guides)

I get messages every day from people asking for help in creating Web sites. This page provides links to some of the excellent resources available on the Web. It's not intended to be a comprehensive list; if that's what you're looking for, try one of the general reference sites listed below. Instead, this page lists some of the best resources for the major areas of Web page design. If you can't find an answer here, take a look at my HTML FAQ (frequently asked questions) at **http://werbach.com/web/htmlfaq.html**, which has answers to some of the most common questions I get asked about Web page design.

General Reference

These sites have large, organized collections of links to information about every aspect of Web page design.

Andrew King's Webreference
(**webreference.com**)
A very attractive, well-organized site, with loads of links and substantial amounts of original content.

Web Developers Virtual Library
(**http://www.stars.com/**)
Thousands of links to resources on Web page design.

HTML Writers' Guild
(**http://www.hwg.org/resources/html/**)
An excellent annotated list of resources.

NCSA HyperNews
(**http://union.ncsa.uiuc.edu:80/HyperNews/get/www/html.html**)
Links to numerous sources of HTML info posted by users.

World Wide Web section of Yahoo
(**http://www.yahoo.com/Computers/World_Wide_Web/**)
Lots and lots of links, but not much filtering or organization.

HTML Guides

World Wide Web Consortium
(**http://www.w3.org/hypertext/WWW/MarkUp/MarkUp.html**)
These documents describe the HTML markup language, which is used to create Web pages as the official specifications

The Bare Bones Guide to HTML
(**http://werbach.com/barebones/**)
A comprehensive yet concise cheat sheet of HTML tags, including Netscape extensions, in common usage.

HTML Documentation by Ian Graham
(**http://www.utoronto.ca/webdocs/HTMLdocs/NewHTML/intro.html**)
An excellent, detailed (but long) tutorial.

Web Design Group HTML Reference
(**http://206.25.187.118/reference/**)
Lots of great information about HTML, including detailed information on HTML 3.2.

Introduction to HTML by Eric Meyer
(**http://www.cwru.edu/help/introHTML/toc.html**)

Beginner's Guide to HTML
(**http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.htm**)
A tutorial from NCSA, the folks who created the Mosaic browser.

Setting Background and Text Colors

Current versions of most major browsers support background, text, and link colors via the body tag, but the colors have to be entered in the form of a hex triplet. Several pages provide tools to help generate the color codes; choose the one that you find most useful.

Background FAQ by Mark Koenen
(**http://www.sci.kun.nl/thalia/guide/color/faq.html**)
A great resource with lots of information about the color tags and links to various tools.

RGB Triplet Chart
(**http://www.phoenix.net/~jacobson/rgb.html)**
This page has a large graphical chart with about 250 colors and their hex triplet equivalents.

InfiNet Color Index
(**http://www.infi.net/wwwimages/colorindex.html**)
Lists about a hundred different colors and their hex triplet equivalents. Each color has a link that shows you what the color looks like as a background.

Colour Selector
(**http://catless.ncl.ac.uk/Lindsay/colours.html**)
Allows you to select colors for background, text, and links from scrolling lists. Shows you how your choices look on screen and provides the HTML code you need to generate those colors.

Dan Schwarzlander's Color Utility
(**http://dans.onanalysis.com/dev/colors/**)
Lets you dynamically enter and test out different color combinations.

Style Guides

Style and design are often overlooked in creating Web pages, but the way you present information has a tremendous impact on the way people respond to your pages.

Kevin Werbach's thoughts on what makes a good home page
(**http://werbach.com/web/page_design.html**)

Jorn's thoughts on HTML style.
(**http://www.mcs.net/~jorn/html/net/style.html**)

Yale C/AIM Style Guide
(**http://info.med.yale.edu/caim/StyleManual_Top.HTML**)
The dean of Web style guides.

World Wide Wide Consortium style guide
(**http://www.w3.org/hypertext/WWW/Provider/Style/**)

Webtechs HTML Validator
(**http://www.webtechs.com/html-val-svc/**)
Lets you check your HTML code to make sure it complies with the official specifications. It is generally a good idea to validate all of your pages before putting them online.

CGI Scripting -- General

Common Gateway Interface (CGI) scripts can be used to perform many powerful functions, including adding forms, guestbooks, and access counters to your pages, as described below. Your ability to use CGI scripts will usually depend on whether your service provider offers access to the cgi-bin directory of the server.

NCSA's overview of CGI
(**http://hoohoo.ncsa.uiuc.edu/cgi/overview.html**)

CGI Programmer's Reference
(**http://www.best.com/~hedlund/cgi-faq/**)

Selena Sol's Public Domain CGI Scripts
(**http://www.eff.org/%7Eerict/Scripts**)

Matt's Script Archive
(http://www.worldwidemart.com/scripts/)

Shareware CGIs by Chris Stevens.
(http://128.172.69.106:8080/cgi-bin/cgis.html)

CGI Resources
(http://www.halcyon.com/sanford/cgi/)

Fill-Out Forms

The sites below offer information on how to write scripts and HTML code to process fill-out forms in your Web pages. This sometimes requires that you put CGI scripts on your server, so you need to check with your service provider to find out whether they support forms.

Carlos's Forms Tutorial
(http://robot0.ge.uiuc.edu/~carlosp/cs317/ft.4-5.html)
A step-by-step introduction to creating forms on the Web.

Instantaneous Introduction to CGI and Forms
(http://kuhttp.cc.ukans.edu/info/forms/forms-intro.html)
Detailed information on how forms work and how to implement them.

Access Counters

Access counters let you see how many people have accessed your page. They are ususally implemented either by using CGI scripts or by scanning the system-wide access log files that your server generates automatically.

Webaudit
(http://www.wishing.com/webaudit/wa.html)
Allows you to count accesses of your pages without using a CGI script.

Webcounter
(http://www.digits.com/)
Another “third party” counter service that doesn't require a script on your server.

Pagecount
(http://www.pagecount.com/)
Yet another “third party” counter.

Several of the sites listed in the CGI section of this page include counter scripts.

Guest Books

Guest books let people who view your pages "sign in" and leave messages for you and others to peruse. You can create a guestbook manually, as I have, by using a form to gather information and adding the responses to your guest book page by hand. If you are able to put CGI scripts on your server, you can create a guestbook that updates automatically.

The World Famous Guestbook
(**http://www.lpage.com/wguestbk/**)

A guestbook script written in Perl
(**http://www.cs.uoregon.edu/~jhobbs/guestbook**)

Several of the sites listed in the CGI section of this page include guestbook scripts.

Image Maps

Clickable image maps send the user to different pages depending on where they click in a graphic. Try these sites for more info.

Clickable imagemaps by Russ Jones
http://www.ora.com/oracom/inet/html.html
An extensive discussion, taken from a book.

How to do Imagemaps
http://www2.ncsu.edu/bae/people/faculty/walker/hotlist/imagemap.html
A more concise tutorial, but with links to other pages where you can find more info on imagemaps.

Frames

Netscape 2.0 supports a new feature called "Frames" that lets you split up the window into independent scrollable panes, each of which can display a different Web page or image.

The Netscape Frames tutorial by Charlton Rose
(**http://www.newbie.net/frames/**)

Java

Java is a programming language that allows you to embed small "applets" in your Web pages.

Gamelan
(**http://www.gamelan.com/index.shtml**)
Probably the premier Java site on the Web. Lots of free applets to try out.

Graphics Collections

These pages contain libraries of public domain graphics and other tools that you can use to spruce up the look of your pages.

Clipart.com
(**http://www.clipart.com/**)
A huge collection of links to free clip art on the Web.

Barry's Clip Art Server
(**http://www.barrysclipart.com/**)
Another large collection.

Rocket Shop
(**http://www.rocketshop.holowww.com/**)
High-quality 3D clip art.

GIF Wizard
(**http://www.raspberryhill.com/gifwizard.html**)
Automatically optimizes your GIF files to reduce file size.

Pixelsite
(**http://www.pixelsite.com/**)
An amazing interactive graphics renderer and some great freeware clipart.

GIF Animation

One of the most popular ways of creating animated graphics on Web pages is through the use of animated GIFs.

GIF Animation Tutorial
(**http://members.aol.com/royalef/gifmake.htm**)

Tru Reality GIF Animation Gallery
(**http://trureality.com/**)

Rose's Animated GIF Library
(**http://www.wanderers.com/rose/animate1.html**)

GIFWorld
(**http://www.gifworld.com/**)

Embedding Sound Files

There are several methods to embed sound files into their pages so that the sound plays automatically when the page is launched.

Crescendo Help Page
(**http://www.liveupdate.com/embed.html**)
Crescendo makes a Netscape plugin to play midi files. This page described how to put these sound files on your pages.

Javascript Sound
(**http://www.webreference.com/javascriptold/javascript/alleluia.au**)
How to use Netscape's Javascript to play sounds.

Javascript

Javascript is a scripting language created by Netscape to create special effects in Web pages. Despite the name, it is not related to Java.

Javascript Resources
(**http://www.netscape.com/comprod/products/navigator/version_2.0/script/script_info/index.html**)
Links to some good tutorials from the Netscape site.

Advertising Your Pages

Everyone wants people to know about their pages. These sites allow you to register your page with various announcement and "what's new" services on the Web.

Submit It!
(http://www.submit-it.com)
Lets you use one form to submit your page to about a dozen different places.

Webcom instructions on how to publicize your site
(**http://www.webcom.com/~webcom/html/publicize.html**)

In general, the best way to figure out how to do things with the Web is to experiment, and to look at what other people have done and how they have done it. Use the "View Source" command in your browser to see how other people have constructed their HTML. Good luck!

Appendix C

Introduction to Computer Security at NIEHS

[illegible]

Purchasing computers at NIEHS:

[illegible]

1. [illegible]
2. [illegible]
3. [illegible]
4. [illegible]

VAX Computer Center Security:

VAX PASSWORD SECURITY - information and recommended procedures:

- [illegible]
- [illegible]

- [illegible] 12 [illegible]
- [illegible] 6 [illegible] 32 [illegible]
- [illegible]
- [illegible]

Recommendations for maintaining passwords:

- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- F[illegible]

Data Protection Policies:

- [illegible]
- [illegible]
- [illegible]
- [illegible]

- [illegible]

Suspicious Actions that should be reported to the CSP, Systems staff or Security Officer:

- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]

BE SUSPICIOUS. Be safe rather than sorry!!!!!

User responsibilities for both Vax and Desktop Computing:

- [illegible]
- [illegible]

Procedures for PC/MAC/Desktop computing:

- [illegible]

- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
  - [illegible]
  - [illegible]

Protection and privacy of user data and sensitive data:

- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]

- [illegible]
- [illegible]
- [illegible]
- [illegible]

- [illegible]
- [illegible]

Recommended for IBM & Clones: F-PROT Version 2.20
Recommended for Macintosh: SAM--Symantec Antivirus for

PHYSICAL SECURITY CONTROLS

- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible] **15** [illegible]
- [illegible]
- [illegible]
- [illegible]

[illegible]

[illegible] **541 2100** [illegible]

[illegible] **29** [illegible] **97**
[illegible]

http://www.niehs.nih.gov/compuref/security/sec_info.htm